Calcul rapide sur les matrices structurées : Les matrices de Hankel.

Nadia Ben Atti

THÈSE

présentée à

L'U.F.R. DES SCIENCES ET TECHNIQUES
DE L'UNIVERSITÉ DE FRANCHE-COMTÉ

pour obtenir le

GRADE DE DOCTEUR DE L'UNIVERSITÉ DE FRANCHE-COMTÉ
Spécialité : Mathématiques et Applications

CALCUL RAPIDE SUR LES MATRICES STRUCTURÉES : LES MATRICES DE HANKEL

par

Nadia BEN ATTI

Soutenue le 28 **Novembre** 2008 **devant la Commission d'Examen**

Présidente :	Marie-Françoise Roy,	Professeur à l'Université de Rennes 1
Directeur de Thèse :	Henri Lombardi,	Maître de Conférence HDR à l'Université de Franche-Comté
Examinateur :	Hassen Oukhaba,	Maître de Conférence HDR à l'Université de Franche-Comté
Rapporteurs :	Dario-Andrea Bini,	Professeur à l'Université de Pise-Italie
	Laureano Gonzalez-Vega,	Professeur à l'Université de Cantabria-Espagne

Table des matières

Liste des algorithmes

Liste des tableaux

Liste des figures

Remerciements

La rédaction de ce manuscrit est l'aboutissement de plusieurs années de travail. Ce travail lui-même n'aurait pu être mené sans l'aide de plusieurs personnes auxquelles je souhaite exprimer ma gratitude.

Voici donc venu le temps de mettre fin à cette thèse, objet des années de travail acharné sous la direction efficace de monsieur Henri Lombardi, Maître de Conférence à l'Université de Franche-Comté-France, qui sous sa direction j'ai pu comprendre et apprendre le métier de chercheur et une certaine éthique de la recherche. Qu'il trouve ici l'expression de ma profonde gratitude et de ma sincère reconnaissance.

Madame Marie-Françoise Roy, Professeur à l'Université de Rennes 1-France, a accepté fort courtoisement d'être présidente du jury. En retour, je tiens à lui exprimer ma gratitude et ma plus profonde admiration.

Je présente mes plus chaleureux remerciements aux rapporteurs de cette dissertation, monsieur Laureano Gonzalez-Vega, Professeur à l'Université de Cantabria-Espagne et monsieur Dario-Andrea Bini, Professeur à l'Université de Pise-Italie, qui ont sûrement passé de longues heures à lire, commenter et corriger ce document.

Je remercie vivement monsieur Hassen Oukhaba, Maître de Conférence à l'Université de Franche-Comté-France, pour la cordialité avec laquelle il a accepté de faire partie de ce jury.

Je tiens à exprimer ma gratitude tout particulièrement à Gema M. Diaz-Toca, Maître de Conférence à l'Université de Murcia-Espagne, pour l'aide qu'elle a toujours accepté de bon coeur de m'accorder. Je lui suis reconnaissante pour les nombreuses discussion productives, ce qui m'a bien permis de surmonter les obstacles. Je la remercie vivement pour avoir eu l'amabilité et la gentillesse de corriger mon travail et d'avoir été disponible à mon égard malgrè ses occupation.

Je la remercie encore d'avoir transformé au fil des années notre collaboration scientifique en une sincère amitié.

Que mes remerciements soient également adressés à monsieur Jounïdi Abdeljaoued, Maître de Conférence à l'Ecole Supérieure des Sciences et Techniques de Tunis(ESSTT), pour m'avoir soutenu et aider dans mes recherches. Que ces quelques mots qui ne sauraient suffire lui disent la gratitude et le respect que j'éprouve en retour.

Je tiens également à remercier :

•Toute l'équipe de Mathématiques de Besançon qui m'a accueilli et beaucoup aidé. Que chaque membre de cette équipe trouve ici l'expression personnelle de ma profonde reconnaissance.

•Catherine, mais aussi Catherine (ne m'en veux pas de t'avoir fait passer après l'autre !), qui s'acquittent de leurs tâches administratives avec le sourire et une efficacité sans reproche.

•Mes collègues de l'Ecole Nationale des Sciences de l'Informatique (ENSI)-Tunis, ses directeurs et ses secrétaires qui m'ont toujours aidé, soutenu et entouré de leur solidarité.

•Tous les enseignants que j'ai eu durant ma longue scolarité, ainsi que les inconnus qui par un mot ou un geste m'ont permis d'achever ce travail.

Je n'oublie pas de signaler l'énorme soutien moral et financier de la part de ma famille lors de la préparation de ma thèse, pour cela je tiens à remercier de tout coeur mon père Hedi et ma mère Néjia ou plutôt Moufida, que je ne pourrai jamais remercier assez de tout ce qu'il m'ont apporté, mon mari Frej et mes filles Farah et Nada, pour leur sacrifices pendant mes nombreuses absences et pour leur amour sans limite.

Que ceux qui, dans mon cher pays et ma chère famille, ont enduré avec moi l'épreuve de l'éloignement, ou ceux qui, dans ce beau pays d'accueil qu'est la France, Franche-Comté, m'ont aidé à supporter les durs moments et à surmonter l'épreuve, en soient chaleureusement et vivement remerciés.

Nadia BEN ATTI

"La vie est un mystère qu'il faut vivre et non un problème à résoudre"

Introduction

De nombreux travaux ont été réalisés sur les algorithmes matriciels et leur complexités.

L'objet de cette thèse est de présenter une contribution à l'amélioration de certains résultats concernant les algorithmes en Algèbre linéaire et plus particulièrement les algorithmes du calcul rapide pour les matrices structurées de différents types(matrices symétriques, matrice de Hankel, Toeplitz), d'essayer d'améliorer ou de généraliser les résultats existants au cas d'un corps commutatif arbitraire et d'étudier la possibilité d'implémenter des algorithmes améliorés ou des nouveaux algorithmes dans un système de calcul formel.

Nous présentons un nouvel algorithme de diagonalisation par blocs des matrices de Hankel, particulièrement efficace.

Notre travail donne aussi une nouvelle méthode pour comprendre le théorème de Frobenius, qui est un résultat délicat concernant la signature des matrices de Hankel. Nous montrons que la connaissance des mineurs principaux dominants de la matrice permet de prévoir la forme que prendra une réduction de la matrice à une forme canonique « diagonale par blocs Hankel-inférieurs ». Nous en déduisons ainsi une preuve élémentaire purement algébrique du théorème de Frobenius dans le cas de matrices de Hankel régulières.

Cet algorithme fonctionne sur un corps arbitraire et doit permettre de généraliser le théorème de Frobenius à d'autres corps que le corps des réels (sur lequel la classe d'équivalence d'une forme quadratique est entièrement caractérisée par sa signature).

Les matrices de passage dans cet algorithme sont triangulaire supérieures et sont fabriquées à partir de matrices du type Toeplitz-supérieures. Nous en déduisons qu'il peut s'interpréter comme une succession de calculs d'inverses de développements limités.

Nous interprètons enfin la version simplifiée de l'algorithme comme un algorithme d'Euclide « avec troncatures judicieusement controlées » si on remplace les développements limités obtenus par leurs polynômes réciproques.

Nous donnons également une étude approfondie de l'algorithme d'Euclide signé et de ses versions matricielles pour les matrices de Hankel et de Bezout associées à un couple de polynômes. Nous expliquons les rapports existant entre différents algorithmes connus dans la littérature.

Dans le cas où la matrice de Hankel correspond à une suite récurrente linéaire, nous trouvons ainsi l'algorithme de Berlekamp-Massey, mais dans une version modifiée, et accélérée par les troncatures.

Ce résultat nous semble particulièrement significatif. Il donne en effet une façon très simple et très géométrique de comprendre l'algorithme de Berlekamp-Massey. Il donne à notre sens « la vraie raison » pour laquelle l'algorithme de Berlekamp-Massey fait ce qu'il promet de faire.

Enfin, et ce n'est pas rien, il s'agit maintenant d'un algorithme général qui s'applique à toutes les matrices de Hankel, qu'elles correspondent ou non à une suite récurrente linéaire. Il permet de calculer avec un petit nombre d'opérations arithmétiques (un peu plus faible que celui qui intervient dans l'algorithme de Berlekamp-Massey) une forme diagonale par blocs Hankel-inférieurs de la matrice de départ.

Nous commençons la Thèse par un "faux" chapitre, consacré à la présentation des différentes

notations et terminologies utilisées dans cette thèse.

Dans le premier chapitre, nous présentons, tout d'abord un algorithme de « diagonalisation par blocs » d'une forme de Hankel arbitraire, dans lequel les blocs seront aussi de type Hankel-inférieures et les matrices de passage successives sont du type Toeplitz-supérieure. Ensuite, nous donnons une version accélérée de cet algorithme. Enfin, nous donnons une nouvelle preuve du Théorème de Frobenius.

Le deuxième chapitre est consacré à l'application de l'algorithme de diagonalisation présenté dans le premier chapitre sur la matrice de Hankel et de Bezout associée à deux polynômes premiers entre eux U et V. Cela mène à une meilleure compréhension du parallélisme entre l'algorithme d'Euclide étendu et la diagonalisation par blocs, montré dans [7]. Nous montrons aussi que le nouvel algorithme induit sur la matrice de Bezout calcule la même forme diagonale que l'algorithme de [7], en s'appuiant sur une factorization bien connue :

$$\text{Bez (U, V)} = \text{Bez (U, 1) H (U, V) Bez (U, 1)}.$$

La technique utilisée pour parvenir à la mise en oeuvre efficace de l'algorithme du chapitre 1 pour la diagonalisation par blocs d'une matrice de Hankel est utilisée dans le chapitre 3 pour concevoir une différente version de l'algorithme de Berlekamp-Massey dans laquelle le nombre d'opérations arithmétiques est réduit par un facteur de 3/4. Cette accélération, bien que modéré, est intéressante du point de vue théorique.

La thèse contient aussi deux annexes avec la mise en oeuvre des codes MAPLE des algorithmes présentés et quelques exemples concrets.

Si nous avons choisi le logiciel de Calcul Formel MAPLE, c'est à cause de la simplicité et de la transparence de ses fonctions de base, de sa modularité, de la richesse de sa bibliothèque et de la souplesse du langage de programmation qui lui est rattaché, très proche de tous les langages classiques, permettant de définir et de présenter de manière lisible et efficace les algorithmes considérés. C'est aussi à cause de la structure des objets qu'il manipule, bien adaptée aux modèles de calcul utilisés.

"Celui qui cherche la sagesse est un sage, celui qui croit l'avoir trouvée est un fou" (Sénèque)

Notations et Préliminaires

Introduction

Le principal contenu de ce chapitre est la présentation des définitions essentielles à ce travail, illustrées par quelques exemples.

Une *matrice de Hankel* est une matrice (pas nécessairement carrée) $H = (v_{ij})$ dont les coefficients sont constants sur les diagonales montantes : $v_{ij} = v_{pq}$ si $i + j = p + q$.

Les matrices de Hankel fournissent un exemple de *matrices structurées*. L'autre exemple le plus important est celui des *matrices de Toeplitz*, celles dont les coefficients sont constants sur les diagonales descendantes : $v_{ij} = v_{pq}$ si $i - j = p - q$.

Remarquons qu'une matrice de Hankel carrée d'ordre n est une matrice *symétrique* et que les produits $H\mathrm{J}_n$ et $\mathrm{J}_n H$ d'une matrice de Hankel H carrée d'ordre n par la matrice de Hankel particulière J_n :

$$\mathrm{J}_n = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ \vdots & \vdots & & \iddots & 0 & 0 \\ \vdots & 0 & \iddots & & \vdots & \vdots \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix},$$

sont des matrices de Toeplitz. Cette matrice de permutation d'ordre n permet de renverser l'ordre des n colonnes (resp. des n lignes) d'une matrice lorsque celle-ci est multipliée à droite (resp.à gauche) par la matrice J_n : c'est pourquoi on l'appelle *matrice de renversement* ou encore *matrice d'arabisation* du fait qu'elle permet d'écrire de droite à gauche les colonnes que l'on lit de gauche à droite et inversement (voir [1]).

Inversement, les produits $T\mathrm{J}_n$ et $\mathrm{J}_n T$ d'une matrice de Toeplitz T carrée d'ordre n par la matrice J_n sont des matrices de Hankel .

Une matrice structurée est déterminée par la donnée de beaucoup moins de coefficients qu'une matrice ordinaire de même taille. Par exemple une matrice de Hankel (resp. de Toeplitz) de type (n, p) est déterminé par la donnée de $n + p - 1$ coefficients : ceux des première ligne et dernière (resp.première) colonne.

Cela rend ces matrices particulièrement importantes pour les « grands calculs » d'algèbre linéaire.

Dans tous ce qui suit nous désignons par :

- $\mathbb{K}$ un corps. Ce sera un corps ordonné lorsqu'on traitera des questions de signes et de signatures.
- Hk(L) la matrice carrée de Hankel d'ordre n définie par la liste L de taille $(2n - 1)$, c'est-à-dire la matrice de Hankel dont la première ligne est constituée des n premiers coefficients de L et dont la dernière colonne est formée par les n derniers coefficients de

L. Par exemple :

$$\mathrm{Hk}(1,2,3,4,5) = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}$$

Notons que pour des raisons de simplification on a écrit $\mathrm{Hk}(1,2,3,4,5)$ au lieu d'écrire $\mathrm{Hk}([1,2,3,4,5])$.

- $\mathrm{Hk}(\mathrm{L};m;n)$ la matrice de Hankel de type (m,n) définie par la liste L de taille $(m+n-1)$, c'est-à-dire la matrice de Hankel dont la première ligne est constituée des m premiers coefficients de L et dont la dernière colonne est formée par les n derniers coefficients de L (un exemple est donné plus bas).
- $\mathrm{Hks}(\mathrm{L})$ la matrice carrée Hankel-supérieure définie par la liste L, c'est-à-dire la matrice de Hankel ayant L pour première ligne et dont la dernière colonne est nulle sauf éventuellement le premier coefficient .
- $\mathrm{Hki}(\mathrm{L})$ la matrice carrée Hankel-inférieure définie par la liste L, c'est-à-dire la matrice de Hankel ayant L pour dernière colonne et dont la première ligne est nulle sauf éventuellement le dernier coefficient (un exemple est donné plus bas).
- $\mathrm{J}_n = \mathrm{Hki}(1,\underbrace{0,\ldots,0}_{(n-1)}), \qquad \forall n \in \mathbb{N}$. Par exemple $\mathrm{J}_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$.
- $\widetilde{P}$ la matrice de type (n,m) définie pour toute matrice P de type (m,n) de la manière suivante : $\widetilde{P} = \mathrm{J}_n\,{}^{\mathrm{t}}P\mathrm{J}_m$.
- $\mathrm{Top}(\mathrm{L})$ la matrice carrée de Toeplitz d'ordre n définie par la liste L de taille $(2n-1)$, c'est-à-dire la matrice de Toeplitz dont la première colonne est constituée (dans le sens ascendant) des n premiers coefficients de L et dont la première ligne est formée par les n derniers coefficients de L. Par exemple :

$$\mathrm{Top}(1,2,3,4,5) = \begin{pmatrix} 3 & 4 & 5 \\ 2 & 3 & 4 \\ 1 & 2 & 3 \end{pmatrix}.$$

- $\mathrm{Top}(\mathrm{L};m;n)$ la matrice de Toeplitz de type (m,n) définie par la liste L de taille $(m+n-1)$, c'est-à-dire la matrice de Toeplitz dont la première colonne est constituée (en montant) par les m premiers coefficients de L et dont la première ligne est formée par les n derniers coefficients de L. Par exemple :

$$\mathrm{Hk}(5,6,7,8,9,10;3;4) = \begin{pmatrix} 5 & 6 & 7 & 8 \\ 6 & 7 & 8 & 9 \\ 7 & 8 & 9 & 10 \end{pmatrix} \quad \text{et} \quad \mathrm{Top}(5,6,7,8,9,10;3;4) = \begin{pmatrix} 7 & 8 & 9 & 10 \\ 6 & 7 & 8 & 9 \\ 5 & 6 & 7 & 8 \end{pmatrix}.$$

  On a $\mathrm{Top}(\mathrm{L};m;n) = \mathrm{J}_m\mathrm{Hk}(\mathrm{L};m;n)$.
- $\mathrm{Topi}(\mathrm{L})$ la matrice Toeplitz-inférieure (triangulaire inférieure de Toeplitz) définie par la liste L, c'est-à-dire la matrice de Toeplitz ayant L pour première colonne et dont la première ligne est nulle sauf éventuellement le premier coefficient.
- $\mathrm{Tops}(\mathrm{L})$ la matrice Toeplitz-supérieure (triangulaire superieure de Toeplitz) définie par la liste L, c'est-à-dire la matrice de Toeplitz ayant L pour première ligne et dont la première colonne est nulle sauf éventuellement le premier coefficient. Par exemple :

$$\mathrm{Hki}(1,2,3) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix} \qquad \text{et} \qquad \mathrm{Tops}(1,2,3) = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}.$$

  On a $\mathrm{Tops}(\mathrm{L}) = \mathrm{J}_\ell\,\mathrm{Hki}(\mathrm{L})$ où ℓ est la longueur de L.

- $\mathrm{Diag}(M_1, M_2, \ldots, M_n)$ la matrice diagonale par bloc de matrices carrées $(M_i)_{1\leq i\leq n}$.
- Pour toute matrice N de type (m, n), nous désignons par N_k la sous matrice (principale) carrée d'ordre k $(1 \leq k \leq \inf(m, n))$ obtenue à partir des k premières lignes et colonnes de N.
- $\Delta_{\mathrm{k}}(N)$ le mineur principal dominant d'ordre k $(1 \leq k \leq \inf(m, n))$ de la matrice N de type (m, n). Par convention $\Delta_0(N) = 1$, et $\Delta_\ell(N) = 0$ si $\ell > \inf(m, n)$.
- Si $P(X) = \sum_{i=s}^{0} p_i X^i \in \mathbb{K}[X]$, alors $\mathbf{P} = (p_s, \ldots, p_0)$.
- Un polynôme formel à coefficients dans $\mathbb{K}$ est un couple $(P(X), n)$ où $P(X) \in \mathbb{K}[X]$ et $n \geq \deg(P(X))$, n est alors appelé le *degré formel* du polynôme formel $(P(X), n)$.
- $P(X)_k$ est le polynôme de degré k obtenu à partir du polynôme $P(X)$ en supprimant les puissances strictement supérieures à k. c'est-à-dire $P(X)_k = P(X) \mod X^{k+1}$.
- $\widehat{P}(X)$ est le polynôme réciproque du polynôme $P(X)$, défini par : $\widehat{P}(X) = X^d P(\frac{1}{X})$ où d est le degré formel de $P(X)$.

1. Diagonalisation par blocs des formes de Hankel

Introduction

Ce chapitre est consacré à la « diagonalisation par blocs Hankel-inférieurs » d'une forme de Hankel arbitraire : la matrice h est remplacée par une matrice $Lh\,{}^{\mathrm{t}}L$ dans laquelle la matrice de passage L est triangulaire inférieure, la forme réduite obtenue étant une matrice diagonale par blocs Hankel-inférieurs. Dans la littérature anglaise, on parlerait plutôt de « block LU-factorization », le mot « block diagonalization » étant réservé à des calculs de matrices par blocs. La terminologie « block LU-factorization » ne nous semble pas très heureuse dans la mesure où ce qu'on a en vue est une forme réduite diagonale $Lh\,{}^{\mathrm{t}}L$, correspondant à la diagonalisation d'une forme quadratique : si L est le L de LU, le U de LU est en fait égal à $h\,{}^{\mathrm{t}}L$. Malgré le risque de confusion avec le « block diagonalization » de la littérature anglaise, et nous nous en excusons auprès des lecteurs, nous utilisons « diagonalisation par blocs » tout court lorsque le contexte montre que ce qui est en vue est une « diagonalisation par blocs Hankel-inférieurs » d'une forme de Hankel. Après un rappel (section 1.1) concernant « la méthode classique » nous présentons (section 1.2) un nouvel algorithme de « diagonalisation par blocs Hankel-inférieurs » d'une forme de Hankel arbitraire, dans lequel les matrices de passage successives sont du type Toeplitz-supérieure. Nous donnons ensuite une version simplifiée et accélérée de notre algorithme (section 1.3) ainsi qu'une variante (section 1.4) qui donne un calcul d'une suite de restes « tronqués » très semblable à l'algorithme d'Euclide (pour le calcul du pgcd de deux polynômes). La section 1.5 présente brièvement la comparaison entre notre méthode et la métode classique. Dans la section 1.6, comme application de la diagonalisation par blocs Hankel-inférieurs nous présentons une preuve algébrique simple du théorème de Frobenius, qui donne la signature d'une matrice de Hankel réelle à partir des signes des mineurs principaux dominants.

Nous tenons à signaler enfin qu'une version abrégée en anglais de ce chapitre a fait l'objet de l'article « Blok Diagonalization and LU-equivalence of Hankel matrices » publié dans la revue *Linear Algebra and its Applications*, (voir [4]).

1.1 Méthode classique : décomposition LU-équivalente d'une matrice de Hankel

Définition 1.1.1. *Soient A et B deux matrices carrées d'ordre n. A et B sont dites LU-équivalentes s'il existe deux autres matrices P et Q telles que P unitriangulaire inférieure et Q unitriangulaire supérieure, vérifiant $A = P\,B\,Q$.*

Dans ([3],[7], [8], [9], [22] et [23]), on trouve le résultat suivant qui assure qu'une matrice de Hankel h est LU-équivalente à une matrice diagonale par blocs. Ci-après nous donnons la version écrite dans [8].

Théorème 1.1. *Soit* $h = \mathrm{Hk}(h_1, h_2, \ldots, h_{2n-1})$ *une matrice de Hankel, régulière d'ordre* n *et* $H_p = \mathrm{Hk}(h_1, h_2, \ldots, h_{2p-1}), \quad p = 1, \ldots, n$, *la sous-matrice principale dominante d'ordre* p *de* h. *Soit* $m_0 = 0$ *et* $1 \leq m_1 < \cdots < m_k = n$ *les ordres des mineurs principaux dominants non nuls de* h. *Posons* $r_i = m_i - m_{i-1}$ *pour* $i = 1, \ldots, k$. *Enfin désignons par* $\mathcal{D}_n$ *la classe des matrices d'ordre* n, *diagonales par blocs* $\mathrm{Diag}(D_{11}, D_{22}, \ldots, D_{kk})$ *où chaque* D_{ii} *est Hankel-inférieure. On a alors les résultats suivants :*
(1) Il existe une matrice unitriangulaire supérieure R *telle que :*

$$^{\mathrm{t}}RhR = D_R \in \mathcal{D}_n \tag{1.1}$$

En particulier, on peut choisir

$$R = R_{\mathrm{class}} = (\mathbf{q}_1, Z_n\mathbf{q}_1, \ldots, Z_n^{r_1-1}\mathbf{q}_1, \mathbf{q}_2, Z_n\mathbf{q}_2, \ldots, Z_n^{r_2-1}\mathbf{q}_2, \ldots, \mathbf{q}_k, Z_n\mathbf{q}_k, \ldots, Z_n^{r_k-1}\mathbf{q}_k),$$

où Z_n *est la matrice Toeplitz inférieure d'ordre* n, $Z_n = \mathrm{Topi}(0, 1, 0, \cdots, 0)$, *de décalage vers le bas (connu en anglais par "down-shift matrix") et*

$$\mathbf{q}_1 = \mathbf{e}_1, \quad \mathbf{q}_i = \begin{pmatrix} s_i \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad s_i = -H_{m_{i-1}}^{-1} \begin{pmatrix} h_{m_{i-1}+1} \\ \vdots \\ h_{2m_{i-1}} \end{pmatrix} \in \mathbb{R}^{m_{i-1}}, \quad \text{pour } i = 2, \ldots, k.$$

(2) Toute matrice unitriangulaire supérieure $M = (\mathrm{col}_1, \mathrm{col}_2, \ldots, \mathrm{col}_n)$ *telle que* $^{\mathrm{t}}MhM \in \mathcal{D}_n$ *vérifie pour* $i = 0, \ldots, k-1$ *: le* $i+1$*-ème bloc est d'ordre* r_{i+1} *et* $\mathrm{col}_{m_i+1} = \mathbf{q}_{i+1}$.
(3) Soit, $Q_i(X) = (1, X, X^2, \ldots, X^{n-1})\mathbf{q}_{i+1}, \quad i = 0, \ldots, k, \quad$, *où* $\mathbf{q}_{k+1} = -H_n^{-1} \begin{pmatrix} h_{n+1} \\ \vdots \\ h_{2n} \end{pmatrix}$, *on a alors :*

$$\begin{aligned} &\deg(Q_i) = m_i, \quad i = 0, \ldots, k, \\ &Q_i(X) = c_i(X)Q_{i-1}(X) - \theta_{i-1}Q_{i-2}(X), \quad i = 1, \ldots, k \\ &Q_0(X) = 1, \quad Q_{-1}(X) = 0, \end{aligned}$$

où $c_i(X)$ *est un polynôme unitaire de degré* r_i *et* θ_{i-1} *une constante, pour* $i = 1, \ldots, k$.

Nous allons présenter dans ce chapitre un algorithme permettant d'avoir une autre diagonalisation par blocs d'une matrice de Hankel, dans laquelle chaque bloc diagonal est aussi Hankel-inférieur, sans utiliser aucune multiplication des matrices, mais seulement des opérations sur les polynômes.

Plus précisément, on décrit une méthode permettant d'obtenir la matrice D pour une diagonalisation par blocs Hankel-inférieurs

$$^{\mathrm{t}}AhA = D \in \mathcal{D}_n.$$

avec une matrice triangulaire supérieure A qu'il n'est pas nécessaire de calculer. Ses éléments diagonaux sont non nuls mais non nécessairement égaux à 1.

À la fin du chapitre et dans la section 1.5, nous donnons un exemple comparatif des deux méthodes.

1.2 Une nouvelle méthode de réduction d'une matrice de Hankel par des matrices triangulaires supérieures de Toeplitz

Dans cette section nous présentons une méthode de réduction d'une forme de Hankel en une matrice diagonale par blocs « Hankel-inférieurs ».

1.2.1 Une étape élémentaire de la réduction

Dans cet exemple, nous montrons la première étape d'une « diagonalisation par blocs » d'une forme de Hankel au moyen d'une matrice de changement de base qui est de Toeplitz-supérieure.

Exemple 1.2.1.
Sur $\mathbb{Z}_p$ considérons la matrice de Hankel d'ordre 6 suivante :

$$h = \mathrm{Hk}(0,0,1,3,2,5,6,4,8,9,7) = \begin{pmatrix} 0&0&1&3&2&5\\ 0&1&3&2&5&6\\ 1&3&2&5&6&4\\ 3&2&5&6&4&8\\ 2&5&6&4&8&9\\ 5&6&4&8&9&7 \end{pmatrix} = \left(\begin{array}{ccc|ccc} 0&0&1&3&2&5\\ 0&1&3&2&5&6\\ 1&3&2&5&6&4\\ \hline 3&2&5&6&4&8\\ 2&5&6&4&8&9\\ 5&6&4&8&9&7 \end{array}\right).$$

Nous construisons à partir de h une matrice Toeplitz-supérieure t :

$$t = \mathrm{Tops}(1,3,2,5,6,4) = \begin{pmatrix} 1&3&2&5&6&4\\ 0&1&3&2&5&6\\ 0&0&1&3&2&5\\ 0&0&0&1&3&2\\ 0&0&0&0&1&3\\ 0&0&0&0&0&1 \end{pmatrix} = \left(\begin{array}{ccc|ccc} 1&3&2&5&6&4\\ 0&1&3&2&5&6\\ 0&0&1&3&2&5\\ \hline 0&0&0&1&3&2\\ 0&0&0&0&1&3\\ 0&0&0&0&0&1 \end{array}\right).$$

Et nous calculons son inverse,

$$t^{-1} = \begin{pmatrix} 1&-3&7&-20&55&-146\\ 0&1&-3&7&-20&55\\ 0&0&1&-3&7&-20\\ 0&0&0&1&-3&7\\ 0&0&0&0&1&-3\\ 0&0&0&0&0&1 \end{pmatrix} = \left(\begin{array}{ccc|ccc} 1&-3&7&-20&55&-146\\ 0&1&-3&7&-20&55\\ 0&0&1&-3&7&-20\\ \hline 0&0&0&1&-3&7\\ 0&0&0&0&1&-3\\ 0&0&0&0&0&1 \end{array}\right).$$

Nous utilisons t^{-1} comme matrice de changement de base pour h, nous obtenons :

$$h' = {}^{\mathrm{t}}t^{-1}ht^{-1} = \left(\begin{array}{ccc|ccc} 0&0&1&0&0&0\\ 0&1&-3&0&0&0\\ 1&-3&7&0&0&0\\ \hline 0&0&0&-55&146&-390\\ 0&0&0&146&-390&1046\\ 0&0&0&-390&1046&-2802 \end{array}\right)$$

$$= \begin{pmatrix} \mathrm{Hki}(1,-3,7) & 0\\ 0 & -\mathrm{Hk}(55,-146,390,-1046,2802) \end{pmatrix} = \begin{pmatrix} h'_{11} & 0\\ 0 & h'_{22} \end{pmatrix}.$$

Remarquons que :

- le premier bloc diagonal est Hankel-inférieur, il n'apparaît pas dans la matrice initiale h et sa signature est simple à calculer.
- le deuxième bloc diagonal est de nouveau de Hankel
- les coefficients de h' sont déterminés de manière exacte si nous inversons la matrice unitriangulaire supérieure de Toeplitz d'ordre 9 suivante :

$$T = \mathrm{Tops}(1,3,2,5,6,4,8,9,7) = \begin{pmatrix} 1 & 3 & 2 & 5 & 6 & 4 & 8 & 9 & 7 \\ 0 & 1 & 3 & 2 & 5 & 6 & 4 & 8 & 9 \\ 0 & 0 & 1 & 3 & 2 & 5 & 6 & 4 & 8 \\ 0 & 0 & 0 & 1 & 3 & 2 & 5 & 6 & 4 \\ 0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \left(\begin{array}{ccc|cccccc} 1 & 3 & 2 & 5 & 6 & 4 & 8 & 9 & 7 \\ 0 & 1 & 3 & 2 & 5 & 6 & 4 & 8 & 9 \\ 0 & 0 & 1 & 3 & 2 & 5 & 6 & 4 & 8 \\ \hline 0 & 0 & 0 & 1 & 3 & 2 & 5 & 6 & 4 \\ 0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 2 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right).$$

Nous trouvons :

$$T^{-1} = \left(\begin{array}{ccc|cccccc} 1 & -3 & 7 & -20 & 55 & -146 & 390 & -1046 & 2802 \\ 0 & 1 & -3 & 7 & -20 & 55 & -146 & 390 & -1046 \\ 0 & 0 & 1 & -3 & 7 & -20 & 55 & -146 & 390 \\ \hline 0 & 0 & 0 & 1 & -3 & 7 & -20 & 55 & -146 \\ 0 & 0 & 0 & 0 & 1 & -3 & 7 & -20 & 55 \\ 0 & 0 & 0 & 0 & 0 & 1 & -3 & 7 & -20 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -3 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right).$$

Nous voyons donc que les coefficients du deuxième bloc Hankel (h'_{22}) sont obtenus à partir des 5 derniers coefficients de la première ligne de T^{-1} à un signe $(-)$ près.

□

Nous allons voir maintenant que l'exemple précédent se généralise.

Soient $n \in \mathbb{N}^*$ et $1 \leq r \leq n-1$. Soit $h = \mathrm{Hk}(\alpha_1, \ldots, \alpha_{2n-1})$ une matrice carrée de Hankel d'ordre n. Supposons que $\alpha_1 = \ldots = \alpha_{r-1} = 0$ et $\alpha_r \neq 0$, alors h est de la forme :

$$h = \mathrm{Hk}(\underbrace{0, \ldots, 0}_{(r-1)}, \alpha_r, \ldots, \alpha_{2n-1}) = \begin{pmatrix} h_{11} & h_{12} \\ {}^t h_{12} & h_{22} \end{pmatrix},$$

où :

- h_{11} est la matrice carrée Hankel-inférieure d'ordre r suivante :

$$h_{11} = \mathrm{Hki}(\alpha_r, \ldots, \alpha_{2r-1}).$$

- h_{22} est la matrice carrée de Hankel d'ordre $(n-r)$ suivante :

$$h_{22} = \mathrm{Hk}(\alpha_{2r+1}, \ldots, \alpha_{2n-1}).$$

- h_{12} est la matrice de Hankel de type $(r, n-r)$ suivante :

$$h_{12} = \mathrm{Hk}(\alpha_{r+1}, \ldots, \alpha_{n+r-1}; r; n-r).$$

À partir de h nous construisons successivement :
1. la matrice carrée d'ordre n Toeplitz-supérieure t,

$$t = \mathrm{Tops}(\alpha_r, \ldots, \alpha_{n+r-1}) = \begin{pmatrix} t_{11} & t_{12} \\ 0 & t_{22} \end{pmatrix},$$

où :

- t_{11} est la matrice carrée d'ordre r Toeplitz-supérieure,

$$t_{11} = \mathrm{Tops}(\alpha_r, \ldots, \alpha_{2r-1}) = \mathrm{J}_r h_{11}, \tag{1.2}$$

- t_{22} est la matrice carrée d'ordre $(n-r)$ Toeplitz-supérieure,

$$t_{22} = \mathrm{Tops}(\alpha_r, \ldots, \alpha_{n-1}),$$

- t_{12} est la matrice de Toeplitz de type $(r, n-r)$

$$t_{12} = \mathrm{Top}(\alpha_{r+1}, \ldots, \alpha_{n+r-1}; r; n-r) = \mathrm{J}_r h_{12}. \tag{1.3}$$

Remarquons que t n'est autre que la matrice Toeplitz-supérieure ayant comme première ligne la $r^{\text{ième}}$ ligne de h.

2. La matrice carrée Hankel-inférieure H d'ordre $(2n-r)$ dont la dernière colonne est formée de la $r^{\text{ième}}$ ligne de h et des $(n-r)$ derniers coefficients de la dernière colonne de h, c'est-à-dire

$$H = \mathrm{Hki}(\alpha_r, \ldots, \alpha_{2n-1}) = \begin{pmatrix} 0 & 0 & H_{13} \\ 0 & h_{11} & h_{12} \\ H_{13} & {}^{\mathrm{t}}h_{12} & h_{22} \end{pmatrix},$$

où

$$H_{13} = \mathrm{J}_{n-r} t_{22}. \tag{1.4}$$

3. La matrice carrée d'ordre $(2n-r)$ Toeplitz-supérieure $T = \mathrm{J}_{2n-r} H$, dont les coefficients de sa première ligne sont ceux de la dernière colonne de H,

$$T = \mathrm{Tops}(\alpha_r, \ldots, \alpha_{2n-1}) = \begin{pmatrix} t_{22} & \widetilde{t_{12}} & T_{13} \\ 0 & t_{11} & t_{12} \\ 0 & 0 & t_{22} \end{pmatrix},$$

où

$$\widetilde{t_{12}} = \mathrm{J}_{n-r}\, {}^{\mathrm{t}}h_{12}, \tag{1.5}$$

$$T_{13} = \mathrm{J}_{n-r} h_{22}. \tag{1.6}$$

Comme t et T sont Toeplitz-supérieures avec un coefficient diagonal non nul, elles sont régulières et chacune a pour inverse une matrice de même type. Soit $\mu_1, \ldots, \mu_{2n-r}$ tel que $T^{-1} = \mathrm{Tops}(\mu_1, \ldots, \mu_{2n-r})$. On a alors les décompositions par blocs suivantes

$$t^{-1} = \mathrm{Tops}(\mu_1, \ldots, \mu_n) = \begin{pmatrix} t_{11}^{-1} & P \\ 0 & t_{22}^{-1} \end{pmatrix},$$

où P est la matrice de Toeplitz de type $(r, n-r)$,

$$P = \mathrm{Top}(\mu_2, \ldots, \mu_n; r; n-r).$$

D'après l'inversion des matrices par blocs et les Equations (1.2) et (1.3), on a respectivement :

$$t_{11}P + t_{12}\ t_{22}^{-1} = 0_{(r,n-r)}, \tag{1.7}$$

$$h_{11}P + h_{12}t_{22}^{-1} = 0_{(r,n-r)}, \tag{1.8}$$

et

$$T^{-1} = \mathrm{Tops}(\mu_1, \ldots, \mu_{2n-r}) = \begin{pmatrix} t_{22}^{-1} & \widetilde{P} & (T^{-1})_{13} \\ 0 & t_{11}^{-1} & P \\ 0 & 0 & t_{22}^{-1} \end{pmatrix},$$

où

$$\begin{aligned} \widetilde{P} &= \mathrm{J}_{n-r}\,{}^{\mathrm{t}}P\mathrm{J}_r, & (1.9)\\ t_{22}\widetilde{P} + \widetilde{t_{12}}t_{11}^{-1} &= 0_{(n-r,r)}, & (1.10)\\ t_{22}(T^{-1})_{13} + \widetilde{t_{12}}P + T_{13}t_{22}^{-1} &= 0_{(n-r,n-r)}. & (1.11) \end{aligned}$$

Notre premier résultat consiste à dire qu'il est seulement nécessaire de calculer T^{-1} dans le but d'obtenir les deux premiers blocs de la décomposition. En utilisant t^{-1} comme matrice de changement de base pour h, nous allons montrer le théorème suivant

Théorème 1.2. *Supposons* $h = \mathrm{Hk}(\underbrace{0, \ldots, 0}_{(r-1)}, \alpha_r, \ldots, \alpha_{2n-1})$, *avec* $\alpha_r \neq 0$, *et posons :*

$$\begin{array}{rclcrcl} t &=& \mathrm{Tops}(\alpha_r, \ldots, \alpha_{n+r-1}), & & t^{-1} &=& \mathrm{Tops}(\mu_1, \ldots, \mu_n),\\ T &=& \mathrm{Tops}(\alpha_r, \ldots, \alpha_{2n-1}), & & T^{-1} &=& \mathrm{Tops}(\mu_1, \ldots, \mu_{2n-r}). \end{array}$$

On a alors :

$$h' = {}^{\mathrm{t}}t^{-1}ht^{-1} = \begin{pmatrix} h'_{11} & 0 \\ 0 & h'_{22} \end{pmatrix}, \tag{1.12}$$

où :

$$h'_{11} = \mathrm{Hki}(\mu_1, \ldots, \mu_r) = \mathrm{J}\,h_{11}^{-1}\,\mathrm{J} \quad \text{et} \quad h'_{22} = -\mathrm{Hk}(\mu_{r+2}, \ldots, \mu_{2n-r}).$$

Démonstration. Calculons le produit ${}^{\mathrm{t}}t^{-1}ht^{-1}$. On commence tout d'abord par le calcul de ht^{-1}

$$ht^{-1} = \begin{pmatrix} h_{11}t_{11}^{-1} & h_{11}P + h_{12}t_{22}^{-1} \\ {}^{\mathrm{t}}h_{12}t_{11}^{-1} & {}^{\mathrm{t}}h_{12}P + h_{22}t_{22}^{-1} \end{pmatrix} = \begin{pmatrix} \mathrm{J}_r & 0 \\ -H_{13}\widetilde{P} & -H_{13}(T^{-1})_{13} \end{pmatrix}.$$

En effet,

$$h_{11}t_{11}^{-1} \overset{(1.2)}{=} \mathrm{J}_r,$$

$$h_{11}P + h_{12}t_{22}^{-1} \overset{(1.8)}{=} 0_{(r,n-r)},$$

$${}^{\mathrm{t}}h_{12}t_{11}^{-1} \overset{(1.5)}{=} \mathrm{J}_{n-r}\widetilde{t_{12}}t_{11}^{-1} \overset{(1.10)}{=} -\mathrm{J}_{n-r}t_{22}\widetilde{P} \overset{(1.4)}{=} -H_{13}\widetilde{P},$$

$${}^{\mathrm{t}}h_{12}P + h_{22}t_{22}^{-1} \overset{(1.5),(1.6)}{=} \mathrm{J}_{n-r}(\widetilde{t_{12}}P + T_{13}t_{22}^{-1}) \overset{(1.11)}{=} -\mathrm{J}_{n-r}t_{22}(T^{-1})_{13} \overset{(1.4)}{=} -H_{13}(T^{-1})_{13}.$$

Par conséquent

$$h' = {}^{\mathrm{t}}t^{-1}ht^{-1} = \begin{pmatrix} {}^{\mathrm{t}}(t_{11}^{-1})\mathrm{J}_r & 0 \\ {}^{\mathrm{t}}P\mathrm{J}_r - {}^{\mathrm{t}}(t_{22}^{-1})H_{13}\widetilde{P} & -{}^{\mathrm{t}}(t_{22}^{-1})H_{13}(T^{-1})_{13} \end{pmatrix}.$$

Mieux encore, comme $\mathrm{J}_r t_{11}^{-1}$ est une matrice de Hankel, elle est donc symétrique et

$$ {}^{\mathrm{t}}(t_{11}^{-1})\mathrm{J}_r = {}^{\mathrm{t}}(\mathrm{J}_r t_{11}^{-1}) = \mathrm{J}_r t_{11}^{-1}, $$

$$ {}^{\mathrm{t}}P\mathrm{J}_r - {}^{\mathrm{t}}(t_{22}^{-1})H_{13}\widetilde{P} \overset{(1.9),(1.4)}{=} (\mathrm{J}_{n-r} - {}^{\mathrm{t}}(t_{22}^{-1})\mathrm{J}_{n-r}t_{22})\widetilde{P}\mathrm{J}_{n-r}(\mathrm{Id}_{n-r} - \mathrm{Id}_{n-r})\widetilde{P} = 0_{(n-r,r)}, $$

$$ {}^{\mathrm{t}}(t_{22}^{-1})H_{13}(T^{-1})_{13} \overset{(1.4)}{=} \mathrm{J}_{n-r}(T^{-1})_{13}. $$

Et on a alors

$$ h' = {}^{\mathrm{t}}t^{-1}ht^{-1} = \begin{pmatrix} \mathrm{J}_r t_{11}^{-1} & 0 \\ 0 & -\mathrm{J}_{n-r}(T^{-1})_{13} \end{pmatrix}. $$

Comme $T^{-1} = \mathrm{Tops}(\mu_1, \ldots, \mu_{2n-r})$, alors

$$ \mathrm{J}_r t_{11}^{-1} = J_r h_{11}^{-1} J_r = \mathrm{Hki}(\mu_1, \ldots, \mu_r) \quad \text{et} \quad -\mathrm{J}_{n-r}(T^{-1})_{13} = -\mathrm{Hk}(\mu_{r+2}, \ldots, \mu_{2n-r}), $$

d'où

$$ h' = {}^{\mathrm{t}}t^{-1}ht^{-1} = \begin{pmatrix} \mathrm{Hki}(\mu_1, \ldots, \mu_r) & 0 \\ 0 & -\mathrm{Hk}(\mu_{r+2}, \ldots, \mu_{2n-r}) \end{pmatrix} = \begin{pmatrix} {h'}_{11} & 0 \\ 0 & {h'}_{22} \end{pmatrix}. $$

□

Si nous itérons l'étape élémentaire prouvée dans le Théoréme 1.2, jusqu'au moment où l'on obtient comme matrice de Hankel restant à traiter une matrice Hankel-inférieure (non nécessairement régulière), nous obtenons ainsi la matrice diagonale par blocs « Hankel-inférieurs » D.
Dans la section suivante nous montrons que pour obtenir D, il n'est pas nécessaire de faire le produit des matrices et ceci est dû aux propriétés des matrices Toeplitz.

1.2.2 L'algorithme complet de la réduction : Algorithme 1.1

Dans cet algorithme, malgré l'affectation $T_1 := T^{-1}$ qui pourrait laisser penser qu'on inverse une matrice Toeplitz-supérieure, on ne manipule que des polynômes de $\mathbb{K}[X]$, car le calcul de l'inverse d'une matrice Toeplitz-supérieure (inférieure) se fait grâce à une inversion dans un anneau de développements limités [8], ce qui revient à une division de 1 par un autre polynôme selon les puissances croissantes et sans tenir compte des puissances trop grandes. Une telle inversion dans l'anneau des développements limités est nettement plus rapide qu'une inversion générale de matrice. Précisément :

Lemme 1.2.1. *Si* $T = \mathrm{Tops}(\alpha_1, \ldots, \alpha_m)$ *avec* $\alpha_1 \neq 0$ *et* $T_1 = T^{-1} = \mathrm{Tops}(\mu_1, \ldots, \mu_m)$,

$$ \text{en posant : } S(X) = \alpha_1 + \alpha_2 X + \ldots + \alpha_m X^{m-1} = \sum_{k=1}^{m} \alpha_k X^{k-1} $$
$$ Q(X) = \mu_1 + \mu_2 X + \ldots + \mu_m X^{m-1} = \sum_{k=1}^{m} \mu_k X^{k-1} $$

on obtient $S(X)Q(X) = 1 \mod X^m$. *Autrement dit* $Q(X)$ *est le quotient dans la division suivant les puissances croissantes de* 1 *par* $S(X)$ *à l'ordre* $(m-1)$ *:* $Q = 1/S \mod X^m$.

On obtient l'algorithme 1.1 page suivante, dans lequel nous notons par $\mathrm{L} \bullet \mathrm{L}'$ la concaténation des listes L et L′.

Algorithme 1.1. ***Algorithme de réduction d'une matrice de Hankel (1)***

Entrée : Une liste non nulle d'éléments du corps $\mathbb{K}$, $S = [\alpha_1, \alpha_2, \ldots, \alpha_{2n-1}]$, qui code une matrice carrée de Hankel h d'ordre n.

Sortie : La liste L des listes des coefficients qui codent les blocs diagonaux « Hankel-inférieurs » de la réduite h' de h, obtenus au moyen d'une réduction par des Toeplitz-supérieures.

Variables locales : m, r, i : compteurs ; $\lambda_1, \ldots, \lambda_{2m-1} \in \mathbb{K}$; l : liste d'éléments de $\mathbb{K}$; T, T_1 : matrices de Toeplitz-supérieures.

m est l'ordre de la matrice de Hankel qui reste à traiter
l est la liste d'éléments de $\mathbb{K}$ qui code cette matrice de Hankel

Début

initialisation

$m := n$; $\lambda_i := \alpha_i,\ i = 1, \ldots, 2m-1$; $l := [\lambda_1, \ldots, \lambda_{2m-1}]$; $L := [\ ]$; $r :=$ l'indice du premier coefficient non nul de l ;

boucle

tant que $r < m$ **faire**

$T := \mathrm{Tops}([\lambda_r, \ldots, \lambda_{2m-1}])$;
$T_1 := T^{-1} := \mathrm{Tops}([\mu_1, \ldots, \mu_{2m-r}])$;
en fait $[\mu_1, \ldots, \mu_{2m-r}]$ n'est pas calculée par une inversion de matrice (lemme 1.2.1)
$L := L \bullet [[\mu_1, \ldots, \mu_r]]$;
$\lambda_i := -\mu_{r+i+1},\ i = 1, \ldots, 2m-2r-1$
$m := m - r$;
$l := [\lambda_1, \ldots, \lambda_{2m-1}]$;
$r :=$ l'indice du premier coefficient non nul de l ; # (si $l = [0, \ldots, 0]$ alors $r := 2m$)

fin tant que

sortie

$L := L \bullet [[\lambda_m, \ldots, \lambda_{2m-1}]]$.
dans la situation générique, r ne prend que la valeur 1 et en fin d'algorithme, L est une
liste de listes à un seul élément, qui représente une matrice diagonale usuelle

Fin.

Complexité de l'algorithme 1.1

Nous faisons quelques études de complexité mais nous nous limitons pour faire les calculs de majoration à la manière usuelle d'exécuter les multiplications et divisons de polynômes à coefficients dans $\mathbb{K}$.

Le nombre d'opération arithmétiques dans la réduction d'une matrice de Hankel d'ordre n, via l'algorithme 1.1, est maximum lorsque les quotients successifs sont tous de degré 1.

Cela donne pour l'étape $k\,(2 \leq k \leq n)$ le calcul de majoration suivant :

- 1 divisions,
- $2k^2 - k$ multiplications,
- $2k^2 - 5k + 3$ soustractions.

Proposition 1.1. *Le nombre d'opérations arithmétiques, lors de l'exécution de l'algorithme 1.1, est majoré par* $4/3\,n^3 - n^2 + 5/3\,n - 2$.

Ainsi la « complexité algébrique de l'algorithme 1.1 », mesurée par le nombre d'opérations arithmétiques dans $\mathbb{K}$ effectuées durant son exécution sur une matrice de Hankel d'ordre n, est un $\mathrm{O}(n^3)$, car une inversion dans l'anneau des développements limités à l'ordre m coûte $\mathrm{O}(m^2)$

opérations arithmétiques. Dans la version accélérée, l'algorithme 1.2 aura une complexité bien meilleure, en $O(n^2)$.

1.3 Simplification de l'algorithme 1.1

On présente un algorithme optimisé pour la réduction des matrices de Hankel qui débouche sur une amélioration de l'algorithme de Berlekamp-Massey (Chapitre 3). Dans la sous-section suivante nous donnons deux exemples de réduction de formes de Hankel, pour chacun desquels nous montrons comment l'algorithme 1.1 peut être accéléré en évitant de calculer en entier les inverses des matrices triangulaires supérieures de Toeplitz successives. La division est arrêtée dès qu'on a obtenu le nouveau bloc Hankel-inférieur. En pratique le nouvel algorithme remplace les inversions successives (dans des anneaux de développements limités) par un calcul du type « suite des quotients et des restes dans l'algorithme d'Euclide » (dans des anneaux de développements limités).

1.3.1 Exemples

Dans le but de ne pas avoir des coefficients de grande taille, tout le calcul (dans les exemples ci-dessous) se fait modulo un nombre premier.

Exemple 1.3.1.
Ce premier exemple consiste à réduire (modulo le nombre premier 101) une matrice de Hankel d'ordre 5, $h = \mathrm{Hk}(S)$ où S est la liste suivante, dans laquelle le dernier coefficient est une indéterminée :

$$S = [1, 3, 2, 5, 6, 4, 8, 9, b]$$

Les blocs diagonaux dans la réduite sont tous d'ordre 1, nous sommes dans la situation générique. Nous suivons l'algorithme 1.1 et nous indiquons au fur et à mesure comment le transformer. Le processus de la réduction se fait en plusieures étapes :

- Première étape

On doit inverser une matrice Toeplitz-supérieure d'ordre 9. Pour cela et d'après le lemme 1.2.1 on doit effectuer la division suivant les puissances croissantes de 1 par

$$S(X) = 1 + 3X + 2X^2 + 5X^3 + 6X^4 + 4X^5 + 8X^6 + 9X^7 + bX^8$$

à l'ordre 8, ce qui donne modulo X^9 :

$$\underbrace{1 - 3X}_{Q_1(X)} + \underbrace{7X^2 - 20X^3 - 46X^4 - 45X^5 - 14X^6 - 36X^7 - (b+19)X^8}_{X^2 B_1(X)}.$$

Remarquons que si on arrête la division au moment où on a calculé $Q_1(X)$, on obtient :

$$\begin{array}{l|l} 1 & S = 1 + 3X + 2X^2 + 5X^3 + 6X^4 + 4X^5 + 8X^6 + 9X^7 + bX^8 \\ \cdots\cdots\cdots\cdots & \underbrace{1 - 3X}_{Q_1(X)} \\ \underbrace{7X^2 + X^3 + 9X^4 + 14X^5 + 4X^6 + 15X^7 + (27-b)X^8}_{X^2 R_1(X)} & \end{array}$$

ce qui correspond à l'égalité

$$1 = S(X)Q_1(X) + X^2 R_1(X) \qquad \mod X^9$$

avec $\deg R_1 = 6$. Ceci implique que $B_1(X)$ peut être calculé en effectuant la division suivant les puissances croissantes de $R_1(X)$ par $S(X)_6$. Autrement dit

$$B_1(X) = \frac{R_1(X)}{S(X)_6} \qquad \mod X^7$$

Mais à l'étape suivante on doit calculer : $-\dfrac{1}{B_1(X)} \mod X^7$.
Il est donc inutile de calculer B_1 pour l'inverser ensuite modulo X^7 ; en effet

$$-\frac{1}{B_1(X)} = -\frac{S(X)_6}{R_1(X)} \qquad \mod X^7.$$

Nous pouvons résumer cette discussion en écrivant la formule :

$$\frac{1}{S(X)} = Q_1(X) + X^2\,\frac{R_1(X)}{S(X)_6} \qquad \mod X^9$$

dans laquelle la division $\frac{R_1(X)}{S(X)_6}$ doit être effectuée seulement modulo X^7.

- Deuxième étape

On doit inverser la matrice Toeplitz-supérieure d'ordre 7, correspondant au developpement limité $-B_1(X)$ à l'ordre 6, pour cela nous effectuons la division suivant les puissances croissantes de $S(X)_6$ par $-R_1(X)$ à l'ordre 6, ce qui donne modulo X^7 :

$$\underbrace{-29+47X}_{Q_2(X)}\underbrace{-13X^2+42X^3+19X^4+4X^5-(33b+5)X^6}_{X^2B_2(X)}.$$

De même, nous remarquons que si on arrête la division au moment où on a calculé $Q_2(X)$, on obtient

$$\begin{array}{l|l} S(X)_6 = 1+3X+2X^2+5X^3+6X^4+4X^5+8X^6 & -R_1 = -7-X-9X^2-14X^3-4X^4-15X^5-(27-b)X^6 \\ \quad\cdots\cdots\cdots\cdots & \hline \underbrace{-29+47X}_{Q_2(X)} \\ \underbrace{-10X^2+22X^3+43X^4-41X^5+(31+29b)X^6}_{X^2R_2(X)} & \end{array}$$

ce qui correspond à une égalité

$$S(X)_6 = -R_1(X)Q_2(X) + X^2R_2(X) \qquad \mod X^7$$

avec $\deg R_2 = 4$. Ceci implique que $B_2(X)$ peut être calculé en effectuant la division suivant les puissances croissantes de $R_2(X)$ par $-R_1(X)_4$. Autrement dit

$$B_2(X) = -\frac{R_2(X)}{R_1(X)_4} \qquad \mod X^5$$

Ce qui nous permet d'écrire :

$$\frac{S(X)_6}{-R_1(X)} = Q_2(X) + X^2\,\frac{R_2(X)}{-R_1(X)_4} \qquad \mod X^7$$

dans laquelle la division $\dfrac{R_2(X)}{-R_1(X)_4}$ doit être effectuée seulement modulo X^5. Ainsi :

$$\frac{1}{S(X)} = Q_1(X) + \frac{X^2}{-Q_2(X) + X^2\,\dfrac{R_2(X)}{R_1(X)_4}} \qquad \mod X^9$$

◦ Troisième étape

On doit inverser la matrice Toeplitz-supérieure d'ordre 5, correspondant au developpement limité $-B_2(X)$ à l'ordre 4, pour cela nous effectuons la division suivant les puissances croissantes de $-R_1(X)_4$ par $-R_2(X)$ à l'ordre 4, ce qui donne modulo X^5 :

$$\underbrace{-31-38X}_{Q_3(X)}\ \underbrace{-36X^2-26X^3+(b+19)X^4}_{X^2B_3(X)}.$$

Si on arrête la division au moment où on a calculé $Q_3(X)$, on obtient

$$\begin{array}{l|l} -R_1(X)_4 = -7-X-9X^2-14X^3-4X^4 & -R_2(X) = 10-22X-43X^2+41X^3-(31+29b)X^4 \\ \quad\cdots\cdots\cdots\cdots & \underbrace{-31-38X}_{Q_3(X)} \\ \underbrace{44X^2+27X^3+(10b-13)X^4}_{X^2R_3(X)} & \end{array}$$

ce qui correspond à une égalité

$$-R_1(X)_4 = -R_2(X)Q_3(X) + X^2R_3(X) \qquad \text{mod } X^5$$

avec $\deg R_3 = 2$. Ceci implique que $B_3(X)$ peut être calculé en effectuant la division suivant les puissances croissantes de $R_3(X)$ par $-R_2(X)_2$. Autrement dit

$$B_3(X) = -\frac{R_3(X)}{R_2(X)_2} \qquad \text{mod } X^3$$

Ce qui nous permet d'écrire :

$$\frac{-R_1(X)_4}{-R_2(X)} = \frac{R_1(X)_4}{R_2(X)} = Q_3(X) + X^2\,\frac{R_3(X)}{-R_2(X)_2} \qquad \text{mod } X^5$$

dans laquelle la division $\dfrac{R_3(X)}{-R_2(X)_2}$ doit être effectuée seulement modulo X^3.

◦ Quatrième étape

On doit inverser la matrice Toeplitz-supérieure d'ordre 3, correspondant au developpement limité $-B_3(X)$ à l'ordre 2, pour cela nous effectuons la division suivant les puissances croissantes de $-R_2(X)_2$ par $-R_3(X)$ à l'ordre 2, ce qui donne modulo X^3 :

$$\underbrace{-14-46X}_{Q_4(X)} + \underbrace{(9-6b)X^2}_{X^2B_4(X)}.$$

Si on arrête la division au moment où on a calculé $Q_4(X)$, on obtient

$$\begin{array}{l|l} -R_2(X)_2 = 10-22X-43X^2 & -R_3(X) = -44-27X-(10b-13)X^2 \\ \quad\cdots\cdots\cdots\cdots & \underbrace{-14-46X}_{Q_4(X)} \\ \underbrace{(8-39b)X^2}_{X^2R_4(X)} & \end{array}$$

ce qui correspond à une égalité

$$-R_2(X)_2 = -R_3(X)Q_4(X) + X^2R_4(X) \qquad \text{mod } X^3$$

avec $\deg R_4 = 0$. Ceci implique que $B_4(X)$ peut être calculé en effectuant la division suivant les puissances croissantes de $R_4(X)$ par $-R_3(X)_0$. Autrement dit

$$B_4(X) = -\frac{R_4(X)}{R_3(X)_0} \qquad \mod X$$

$$\frac{-R_2(X)_2}{-R_3(X)} = \frac{R_2(X)_2}{R_3(X)} Q_4(X) + X^2 \, \frac{R_4(X)}{-R_3(X)_0} \qquad \mod X^3$$

dans laquelle la division $\dfrac{R_4(X)}{-R_3(X)_0}$ doit être effectuée seulement modulo X.

◦ Dernière étape

On doit calculer le dernier coefficient de la réduite, ce qui correspond à la division suivant les puissances croissante de $-R_4(X)$ par $-R_3(X)_0$ à l'ordre 0, ce qui donne le coefficient $6b - 9$.

◦ Conclusion

On vient dans cet exemple de calculer une suite de quotients

$$Q_1 = 1 - 3X, \; Q_2 = -29 + 47X, \; Q_3 = -31 - 38X, \; Q_4 = -14 - 46X$$

et le dernier coefficient $R = 6b - 9$. Dans cet exemple, la réduite D de h est donnée par les monômes constants de chaque Q_i et R. Ainsi D est de la forme :

$$D = \mathrm{Diag}(1, -29, -31, -14, 6b - 9) = \begin{pmatrix} (1) & & & & \\ & (-29) & & & \\ & & (-31) & & \\ & & & (-14) & \\ & & & & (6b-9) \end{pmatrix}.$$

Et par ailleurs on obtient l'identité suivante :

$$\frac{1}{S(X)} = (1-3X) + \cfrac{X^2}{-(-29+47X) + \cfrac{X^2}{(-31-38X) + \cfrac{X^2}{-(-14-46X) + (6b-9)X^2}}} \qquad \mod X^9 \tag{1.13}$$

c'est-à-dire encore :

$$\frac{1}{S(X)} = Q_1(X) + \cfrac{X^{d_1+d_2}}{-Q_2(X) + \cfrac{X^{d_2+d_3}}{Q_3(X) + \cfrac{X^{d_3+d_4}}{-Q_4(X) + RX^{\mu}}}} \qquad \mod X^N \tag{1.14}$$

avec $d_1 + d_2 + d_3 + d_4 + d_R = n = 5$, $\deg(X^{\mu}R) + d_1 + 2d_2 + 2d_3 + d_4 = N - 1$, $N = 2n - 1$. Ici $d_i = \deg Q_i$, mais dans l'exemple suivant nous verrons qu'il faut prendre pour d_i le degré formel de Q_i.

□

Exemple 1.3.2.

Ce deuxième exemple consiste à réduire (modulo le nombre premier 101), via l'algorithme précédent, une matrice de Hankel d'ordre $n = m_1 = 9$, $h = \mathrm{Hk}(S)$ où S est la liste suivante :

$$S := [0, -50, 0, -26, 0, 34, 0, 8, 1, 20, -9, 40, 19, 8, -7, a, b]$$

Cette fois-ci, les blocs diagonaux dans la réduite sont de tailles variables, ce qui correspond au fait qu'il y a des chutes de degrés irrégulières dans la suite des restes. Comme les quotients n'ont pas toujours le degré attendu nous utilisons la notion de quotient formel (avec un degré formel qui peut être plus grand que le degré effectif). Nous avons cette fois-ci pris pour les deux derniers coefficients une indéterminée.
Le processus de la réduction se fait en plusieures étapes :

- Première étape

Comme $r_1 = 2 < 9 = m_1$, il ne s'agit pas d'une matrice Hankel-inférieure et on doit faire une première réduction. Pour cela et théoriquement on doit inverser une matrice Toeplitz-supérieure d'ordre 16 afin d'obtenir le premier bloc diagonal Hankel-inférieur D_{11} d'ordre 2. D'après le lemme 1.2.1 on doit effectuer la division suivant les puissances croissantes de 1 par

$$S(X) = -50-26X^2+34X^4+8X^6+X^7+20X^8-9X^9+40X^{10}+19X^{11}+8X^{12}-7X^{13}+aX^{14}+bX^{15},$$

à l'ordre 15, avec un quotient formel de degré $d_1 = r = 2$, ce qui donne modulo X^{16} :

$$\underbrace{2+3X^2}_{Q_1(X)} + \underbrace{20X^4 - 4X^6 - 4X^7 + 21X^8 + 24X^9 - 44X^{10} + 44X^{11} + 23X^{12} - 8X^{13} - (4a-49)X^{14} - (20+4b)X^{15}}_{X^4B_1(X)}.$$

Comme

$$D_{11} = \mathrm{Hki}(\mu_1, \mu_{r_1}) = \mathrm{Hki}(\mu_1, \mu_2) = \mathrm{Hki}(2, 0),$$

le polynôme

$$Q_1(X) = \mu_1 + \mu_2 X + \mu_3 X^2$$

définit le premier bloc diagonal Hankel-inférieur D_{11}. De plus, comme $\mu_{r_1+2} = \mu_4 = 0$, $\mu_{r_1+3} = \mu_5 \neq 0$ et $r_1 + 3 < m_1 + 1$, alors on aura une autre étape de réduction et le bloc diagonal Hankel-inférieur D_{22} de l'étape suivante sera d'ordre 2. Remarquons que si on arrête la division au moment où on a calculé $Q_1(X)$, on obtient l'égalité :

$$1 = S(X)Q_1(X) + X^4R_1(X) \qquad \mod X^{16},$$

où

$$R_1(X) = 10-17X^2-2X^3+37X^4+15X^5-39X^6-11X^7-35X^8-43X^9-(24+2a)X^{10}+(21-2b)X^{11}.$$

Ceci implique que $B_1(X)$ pourrait être calculé en effectuant la division suivant les puissances croissantes de $R_1(X)$ par $S(X)_{11}$. Autrement dit

$$B_1(X) = \frac{R_1(X)}{S(X)_{11}} \qquad \mod X^{12}.$$

Mais à l'étape suivante on doit calculer : $-\dfrac{1}{B_1(X)} \mod X^{12}$ pour avoir le deuxième bloc diagonal Hankel-inférieur.
Il est donc inutile de calculer B_1 pour l'inverser ensuite modulo X^{12} ; en effet

$$-\frac{1}{B_1(X)} = -\frac{S(X)_{11}}{R_1(X)} \qquad \mod X^{12}.$$

Ainsi, le calcul de $Q_1(X)$ et de $R_1(X)$ détermine les données de l'étape suivante, en effet :

- l'ordre de la nouvelle matrice de Hankel à réduire ($h := h'_{22}$) sera égal à l'ordre de la matrice de Hankel de cette étape moins l'ordre de ce premier bloc diagonal Hankel-inférieur D_{11} définit par $Q_1(X)$, c'est-à-dire la matrice de Hankel de l'étape suivante sera d'ordre $7 = 9 - 2$ ce qui permet d'écrire l'affectation $m_2 := m_1 - r_1$.

- La différence entre la valuation de $R_1(X)$ ($s_1 = 4$) et le degré formel de $Q_1(X)$ détermine le nombre de zéros au début de la liste qui définit la nouvelle matrice de Hankel et par conséquent l'ordre du nouveau bloc diagonal Hankel-inférieur D_{22} s'il existe ! (si la nouvelle matrice de Hankel n'est pas Hankel-inférieure), ce qui permet d'écrire l'affectation $r_2 := s_1 - r_1$.

Ainsi, à la suite de cette première étape on a une deuxième étape de réduction car ($r_2 = 2 < 7 = m_2$) et nous pouvons résumer cette discussion en écrivant la formule :

$$\frac{1}{S(X)} = Q_1(X) + X^4 \, \frac{R_1(X)}{S(X)_{11}} \qquad \mod \ X^{16},$$

dans laquelle $Q_1(X)$ définit le premier bloc diagonal Hankel-inférieur D_{11} et la division $\dfrac{R_1(X)}{S(X)_{11}}$ doit être effectuée seulement modulo X^{12}.

- Deuxième étape

Le calcul de la première étape donne $r_2 = 2 < 7 = m_2$ et il ne s'agit pas d'une matrice Hankel-inférieure, d'où on doit faire une deuxième réduction. Pour cela et théoriquement on doit inverser une matrice Toeplitz-supérieure d'ordre 12, correspondant au developpement limité $-B_1(X)$ à l'ordre 11 afin d'obtenir le deuxième bloc diagonal Hankel-inférieur D_{22} d'ordre 2, pour cela nous effectuons la division suivant les puissances croissantes de $S(X)_{11}$ par $-R_1(X)$ à l'ordre 11, avec un quotient formel de degré $d_2 = r_2 = 2$, ce qui donne modulo X^{12} :

$$\underbrace{5 + X^2}_{Q_2(X)} + \underbrace{X^3 - 46X^5 - 5X^6 + 23X^7 - 20X^8 - 37X^9 + aX^{10} + (25 + b)X^{11}}_{X^3 B_2(X)}.$$

Comme dans l'étape précédente, et puisque

$$D_{22} = \mathrm{Hki}(\mu_1, \mu_{r_2}) = \mathrm{Hki}(\mu_1, \mu_2) = \mathrm{Hki}(5, 0),$$

le polynôme

$$Q_2(X) = \mu_1 + \mu_2 X + \mu_3 X^2$$

définit le deuxième bloc diagonal Hankel-inférieur D_{22}. De plus, comme on a ici $\mu_{r_2+2} = \mu_4 = 1 \neq 0$ et $r_2 + 2 < m_2 + 1$, alors on aura une autre étape de réduction et le bloc diagonal Hankel-inférieur D_{33} de l'étape suivante sera d'ordre 1. Remarquons que si on arrête la division au moment où on a calculé $Q_2(X)$, on obtient l'égalité :

$$S(X)_{11} = -R_1(X)Q_2(X) + X^3 R_2(X) \qquad \mod \ X^{12},$$

où

$$R_2(X) = -10 - 28X^2 - 49X^3 - 39X^4 + 8X^5 - 33X^6 - (14 + 10a)X^7 - (20 + 10b)X^8.$$

Ceci implique que $B_2(X)$ pourrait être calculé en effectuant la division suivant les puissances croissantes de $R_2(X)$ par $-R_1(X)_8$. Autrement dit

$$B_2(X) = -\frac{R_2(X)}{R_1(X)_8} \qquad \mod \ X^9$$

Mais à l'étape suivante on doit calculer $-\dfrac{1}{B_2(X)} \mod X^9$, pour avoir le troisième bloc diagonal Hankel-inférieur. Il est donc inutile de calculer B_2 pour l'inverser ensuite modulo X^9. Ainsi, le calcul de $Q_2(X)$ et de $R_2(X)$ détermine les données de l'étape suivante, en effet :

- l'ordre de la nouvelle matrice de Hankel à réduire sera égal à l'ordre de la matrice de Hankel de cette étape moins l'ordre de ce deuxième bloc diagonal Hankel-inférieur D_{22} définit par $Q_2(X)$, c'est-à-dire la matrice de Hankel de l'étape suivante sera d'ordre $5 = 7 - 2$ ce qui permet d'écrire l'affectation $m_3 := m_2 - r_2$.
- La différence entre la valuation de $R_2(X)$ ($s_2 = 3$) et le degré formel de $Q_2(X)$ détermine le nombre de zéros au début de la liste qui définit la nouvelle matrice de Hankel et par conséquent l'ordre du nouveau bloc diagonal Hankel-inférieur D_{33} (si la nouvelle matrice de Hankel n'est pas Hankel-inférieure), ce qui permet d'écrire l'affectation $r_3 := s_2 - r_2$.

Ainsi, à la suite de cette deuxième étape on a une troisième étape de réduction car ($r_3 = 1 < 5 = m_3$) et nous pouvons résumer cette discussion en écrivant la formule :

$$\frac{1}{S(X)} = Q_1(X) + \frac{X^4}{-Q_2(X) + X^3 \dfrac{R_2(X)}{R_1(X)_8}} \qquad \mod X^{16}$$

dans laquelle $Q_1(X)$ et $Q_2(X)$ définissent respectivement le premier et le deuxième bloc de D et la division $\dfrac{R_2(X)}{R_1(X)_8}$ doit être effectuée seulement modulo X^9.

- Troisième étape

Comme $r_3 = 1 < 5 = m_3$, il ne s'agit pas d'une matrice Hankel-inférieure, d'où on doit faire une troisième réduction. Pour cela et théoriquement on doit inverser une matrice Toeplitz-supérieure d'ordre 9, correspondant au developpement limité $-B_2(X)$ à l'ordre 8, pour cela nous effectuons la division suivant les puissances croissantes de $-R_1(X)_8$ par $-R_2(X)$ à l'ordre 8, avec pour quotient un polynôme formel de degré $d_3 = r_3 = 1$ ce qui donne modulo X^9 :

$$\underbrace{-1}_{Q_3(X)} \underbrace{-46X^2 - 5X^3 + 28X^4 + 25X^5 - 39X^6 - (20-a)X^7 - (50-b)X^8}_{X^2 B_3(X)}$$

Comme dans l'étape précédente, et puisque

$$D_{33} = \mathrm{Hki}(\mu_{r_3}) = \mathrm{Hki}(\mu_1) = \mathrm{Hki}(-1),$$

le polynôme

$$Q_3(X) = \mu_1$$

définit le troisième bloc diagonal Hankel-inférieur D_{33}. Remarquons que $\deg Q_3(X) = 0 < 1 =$ ordre de D_{33}. De plus on a ici $\mu_{r_3+2} = \mu_3 = -46 \neq 0$ et $r_3 + 2 < m_3 + 1$, par conséquent on aura une autre étape de réduction et le bloc diagonal Hankel-inférieur D_{44} de l'étape suivante sera d'ordre 1.

Si on arrête la division au moment où on a calculé $Q_3(X)$, on obtient

$$-R_1(X)_8 = -R_2(X)Q_3(X) + X^2 R_3(X) \qquad \mod X^9$$

où

$$R_3(X) = 45 - 50X + 2X^2 - 23X^3 - 29X^4 + (25 + 10a)X^5 - (46 - 10b)X^6.$$

Ceci implique que $B_3(X)$ pourrait être calculé en effectuant la division suivant les puissances croissantes de $R_3(X)$ par $-R_2(X)_6$. Autrement dit

$$B_3(X) = -\frac{R_3(X)}{R_2(X)_6} \qquad \mod X^7$$

Ainsi, le calcul de $Q_3(X)$ et de $R_3(X)$ détermine les données de l'étape suivante, en effet :

- la nouvelle matrice de Hankel à réduire sera d'ordre $m_4 := m_3 - r_3$, c'est-à-dire à l'étape suivante on aura une matrice de Hankel d'ordre $4 = 5 - 1$.
- la différence entre la valuation de $R_3(X)$ ($s_3 = 2$) et le degré formel de $Q_3(X)$ détermine l'ordre du nouveau bloc diagonal Hankel-inférieur D_{44} (si la nouvelle matrice de Hankel n'est pas Hankel-inférieure), ce qui permet d'écrire l'affectation $r_4 := s_3 - r_3$.

Ainsi, à la suite de cette troisième étape on a une quatrième étape de réduction car ($r_4 = 1 < 4 = m_4$) et nous pouvons résumer cette discussion en écrivant la formule :

$$\frac{1}{S(X)} = Q_1(X) + \cfrac{X^4}{-Q_2(X) + \cfrac{X^3}{Q_3(X) + X^2 \, \cfrac{R_3(X)}{-R_2(X)_6}}} \qquad \mod \ X^{16}$$

dans laquelle $Q_1(X)$, $Q_2(X)$ et $Q_3(X)$ définissent respectivement le premier, le deuxième et le troisième bloc de D et la division $\dfrac{R_3(X)}{-R_2(X)_6}$ doit être effectuée seulement modulo X^7.

- Quatrième étape

Comme $r_4 = 1 < 4 = m_4$, il ne s'agit pas d'une matrice Hankel-inférieure, d'où on doit faire une quatrième réduction. Pour cela et théoriquement on doit inverser la matrice Toeplitz-supérieure d'ordre 7, correspondant au developpement limité $-B_3(X)$ à l'ordre 6, pour cela nous effectuons la division suivant les puissances croissantes de $-R_2(X)_6$ par $-R_3(X)$ à l'ordre 6, avec pour quotient un polynôme formel de degré $d_4 = r_4 = 1$ ce qui donne modulo X^7 :

$$\underbrace{11 + X}_{Q_4(X)} + \underbrace{(20a - 21)X^5 + (36 + 22a + 20b)X^6}_{X^5 B_4(X)}$$

Comme dans les étapes précédentes, et puisque

$$D_{44} = \mathrm{Hki}(\mu_{r_4}) = \mathrm{Hki}(\mu_1) = \mathrm{Hki}(11),$$

le polynôme

$$Q_4(X) = \mu_1 + \mu_2 X$$

définit le quatrième bloc diagonal Hankel-inférieur D_{44}. De plus, on a ici $\mu_{r_4+2} = \mu_3 = \mu_4 = \mu_5 = 0$, $\mu_6 = \mu_{r_4+5} = 20a - 21 \neq 0$ et $r_4 + 5 > m_4 + 1$, ce qui correspond bien à une matrice Hankel-inférieur d'ordre 3 et on peut conclure qu'on est à la dernière étape.
Si on arrête la division au moment où on a calculé $Q_4(X)$, on obtient

$$-R_2(X)_6 = -R_3(X)Q_4(X) + X^5 R_4(X) \qquad \mod \ X^7$$

où $R_4(X) = 36 + 9a - (44 - 10a - 9b)X$.
Ainsi, le calcul de $Q_4(X)$ et de $R_4(X)$ détermine les données de l'étape suivante, en effet :

- la nouvelle matrice de Hankel à réduire serait d'ordre $m_5 := m_4 - r_4$, c'est-à-dire à l'étape suivante on aura une matrice de Hankel d'ordre $3 = 4 - 1$,
- La différence entre la valuation de $R_4(X)$ ($s_4 = 5$) et le degré formel de $Q_4(X)$ détermine le nombre de zéros au début de la liste qui définit cette nouvelle matrice de Hankel et par conséquent l'ordre du nouveau bloc diagonal Hankel-inférieur D_{55} (si la matrice en cours n'est pas Hankel-inférieure) qui serait égal à $4 = 5 - 1$ strictement plus grand que l'ordre de la matrice de Hankel à réduire, ce qui signifie qu'en fait que la nouvelle matrice est Hankel-inférieure.

Signalons que dans les étapes pécédentes, il est inutile de calculer les polynômes $B_i(X)$, $i = 1, 2, 3$, alors que dans cette étape on doit calculer $B_4(X)$ afin d'avoir les coefficients du dernier bloc (d'ordre 3) de D. C'est ce qu'on va décrire dans la dernière étape.

- Dernière étape

On a ici une matrice Hankel-inférieure (dernier bloc de D) d'ordre 3 et théoriquement ces coefficients sont à un signe $(-)$ près ceux de $B_4(X)$. Le calcul de la quatrième étape implique que $-B_4(X)$ peut être calculé en effectuant la division suivant les puissances croissantes de $-R_4(X)$ par $-R_3(X)_1$ à l'ordre 1, avec pour quotient un polynôme formel de degré 3. Autrement dit

$$B_4(X) = \frac{-R_4(X)}{-R_3(X)_1} \;=\; \frac{R_4(X)}{R_3(X)_1} \qquad \mod\ X^2$$

ce qui donne modulo X^2 le quotient :

$$-B_4(X) = R(X) = 21 - 20a - (36 + 22a + 20b)X.$$

Nous pouvons résumer cette discussion en écrivant la formule finale suivante :

$$\frac{1}{S(X)} = Q_1(X) + \cfrac{X^4}{-Q_2(X) + \cfrac{X^3}{Q_3(X) + \cfrac{X^2}{-Q_4(X) + X^5 R(X)}}} \qquad \mod\ X^{16}, \qquad (1.15)$$

- Conclusion

On vient dans cet exemple de calculer une suite de quotients, qui sont les polynômes formels suivants :

$$\begin{array}{rclrcl} (Q_1, d_1) & = & (2 + 3X^2, 2), & (Q_2, d_2) & = & (5 + X^2, 2), \quad (Q_3, d_3) = (-1, 1) \\ (Q_4, d_4) & = & (11 + X, 1), & (R, d_R) & = & (21 - 20a - (36 + 22a + 20b)X, 3); \end{array}$$

$$d_R = 3 \ \text{ pour avoir } \ d_1 + d_2 + d_3 + d_4 + d_R \;=\; n \;=\; 9$$

Ainsi, pour $1 \leq i \leq 4$, si $Q_i(X) = c_0 + \ldots + c_{d_i} X^{d_i}$, où d_i désigne le degré formel de Q_i ($d_i = r$ à la $i^{\text{ème}}$ étape), alors on a :

$$D_{ii} = \text{Hki}(c_0, \ldots, c_{d_i - 1}),$$

et pour $R(X) = 21 - 20a - (36 + 22a + 20b)\, X + 0\, X^2 + 0\, X^3$, le dernier bloc Hankel-inférieur D_{55}, est donné par :

$$D_{55} = \text{Hki}(0, 21 - 20a, -36 - 22a - 20b).$$

Ainsi et quelque soit les valeurs de a et b, D est de la forme :

$$D = \begin{pmatrix} \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix} & & & & & \\ & \begin{pmatrix} 0 & 5 \\ 5 & 0 \end{pmatrix} & & & & \\ & & (-1) & & & \\ & & & (11) & & \\ & & & & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 21 - 20a \\ 0 & 21 - 20a & -36 - 22a - 20b \end{pmatrix} \end{pmatrix}.$$

Rappelons que le calcul qui a été fait correspond à la formule ci-dessous, dans laquelle on désigne par d_i le degré formel de Q_i $(1 \leq i \leq 4)$:

$$\frac{1}{S(X)} = (2+3X^2) + \cfrac{X^4}{-(5+X^2) + \cfrac{X^3}{-1 + \cfrac{X^2}{-(11+X) + (21-20a-(36+22a+20b)X)X^5}}} \mod X^{16}, \tag{1.16}$$

c'est-à-dire encore, avec $n = 9, d_1 = r = 2, d_2 = 2, d_3 = 1, d_4 = 1, N = 2n - r = 16$:

$$\frac{1}{S(X)} = Q_1(X) + \cfrac{X^{d_1+d_2}}{-Q_2(X) + \cfrac{X^{d_2+d_3}}{Q_3(X) + \cfrac{X^{d_3+d_4}}{-Q_4(X) + R(X)X^\mu}}} \mod X^N, \tag{1.17}$$

Remarquons que :

- si (modulo 101), $21 - 20a \neq 0$ et $-36 - 22a - 20b \neq 0$ (c'est-à-dire $a \neq -4$ et $b \neq 43$) alors : $d_1+d_2+d_3+d_4+d_R = n$, $\deg(X^\mu R)+d_1+2d_2+2d_3+d_4 = N-1$ et par conséquent $\mu = 5$.

- si (modulo 101), $21 - 20a = 0$ et $-36 - 22a - 20b \neq 0$ (c'est-à-dire $a = -4$ et $b \neq 43$) , on aura la formule suivante :

$$\frac{1}{S(X)} = (2+3X^2)+\cfrac{X^4}{-(5+X^2) + \cfrac{X^3}{-1 + \cfrac{X^2}{-(11+X) + (-49-20b)X^6}}} \mod X^{16}, \tag{1.18}$$

  c'est-à-dire encore :

$$\frac{1}{S(X)} = Q_1(X) + \cfrac{X^{d_1+d_2}}{-Q_2(X) + \cfrac{X^{d_2+d_3}}{Q_3(X) + \cfrac{X^{d_3+d_4}}{-Q_4(X) + R(X)X^\mu}}} \mod X^N, \tag{1.19}$$

  et comme précédement $d_1+d_2+d_3+d_4+d_R = n$, $\deg(X^\mu R)+d_1+2d_2+2d_3+d_4 = N-1$ et alors $\mu = 6$.

- si $21 - 20a = -36 - 22a - 20b = 0 \mod 101$ (c'est-à-dire $a = -4$ et $b = 43$) on aura alors :

$$\frac{1}{S(X)} = Q_1(X) + \cfrac{X^{d_1+d_2}}{-Q_2(X) + \cfrac{X^{d_2+d_3}}{Q_3(X) + \cfrac{X^{d_3+d_4}}{-Q_4(X)}}} \mod X^N. \tag{1.20}$$

□

1.3.2 Le résultat général

On vient de voir dans les Exemples 1.3.1 et 1.3.2 qu'à chaque étape i de la réduction, on peut éviter de calculer B_i en entier, en effet on arrête la division suivant les puissances croissantes au moment où on a calculé Q_i, le quotient formel nécessaire.

Ainsi, à la matrice carrée de Hankel, $h = \mathrm{Hk}(\alpha_1, \ldots, \alpha_{2n-1})$ avec $\alpha_1 = \ldots = \alpha_{r-1} = 0$, $\alpha_r \neq 0$, on associe le polynôme de degré $2n-r-1$: $S(X) = \sum_{i=r}^{2n-1} \alpha_i X^{i-r}$ et on suit la procédure décrite dans les exemples précédents. On calcule les polynômes formels $(Q_1, d_1), \ldots, (Q_k, d_k)$ et (R, d_R) avec $\sum_{i=1}^{k} d_i + d_R = n$. On a alors le résultat suivant :

Proposition 1.2. *Chaque polynôme formel (Q_i, d_i), $Q_i(X) = c_{0,i} + \ldots + c_{d_i,i}X^{d_i}$, code le $i^{\text{ème}}$ bloc Hankel-inférieur D_{ii} d'ordre d_i de la manière suivante :*

$$D_{ii} = \mathrm{Hki}(c_{0,i}, \ldots, c_{d_i-1,i}).$$

Et en ce qui concerne le dernier bloc $D_{k+1,k+1}$, si $R(X) = r_0 + \ldots + r_t X^t + \ldots + r_{d_R}X^{d_R}$ avec $r_t \neq 0$ et $r_i = 0$ pour $i > t$, alors :

$$D_{k+1,k+1} = \begin{cases} \mathrm{Hki}(r_0, \ldots, r_{d_R-1}) & \text{si } t = d_R \\ \mathrm{Hki}(\underbrace{0, \ldots, 0}_{d_R-t-1}, r_0, \ldots, r_t) & \text{si } t < d_R \end{cases}$$

Mieux encore on a :

$$\frac{1}{S(X)} = Q_1(X) + \cfrac{X^{d_1+d_2}}{-Q_2(X) + \cfrac{X^{d_2+d_3}}{\ddots + \cfrac{\cdots}{(-1)^k Q_{k-1}(X) + \cfrac{X^{d_{k-1}+d_k}}{(-1)^{k+1}Q_k(X) + R(X)X^{\mu}}}}} \mod X^N \tag{1.21}$$

avec $N = 2n - r$, et si $R \neq 0$ alors $\deg(X^{\mu}R) + d_1 + 2d_2 + \cdots + 2d_{k-1} + d_k = N - 1$.

Comme conséquence on obtient l'algorithme 1.2.

1.3.3 Algorithme 1.2

Dans l'algorithme simplifié on utilise la procédure QuoResCroiss(R_0, R_1, r, p). Elle prend en entrée deux polynômes formels R_0 et R_1 (supposés avoir leur coefficient constant non nul et être de même degré formel p) modulo X^{p+1}. La procédure retourne $[Q, R_2]$ où Q est le quotient de degré formel r ($r < p$) et R_2 est le reste, lorsqu'on effectue la division en puissances croissantes de R_0 par R_1 modulo X^{p+1}.

Autrement dit on a dans l'anneau des développements limités à l'ordre p : $R_0 = Q \cdot R_1 + R_2$ avec $\deg Q \leq r$ et $\mathrm{val}\, R_2 > r$. On obtient alors l'algorithme 1.2 page suivante dont la sortie est une liste $Stu = [Stu[1], Stu[2], \ldots, Stu[j]]$, chacun des $Stu[i]$ est un polynôme formel sur $\mathbb{K}[X]$, qui code le $i^{\text{ème}}$ bloc diagonal « Hankel-inférieur » D_{ii}, de la réduite D de h.

Algorithme 1.2. ***Algorithme de réduction d'une matrice de Hankel(2)***

Entrée : Une liste non nulle d'éléments du corps $\mathbb{K}$, $S = [\alpha_1, \alpha_2, \ldots, \alpha_{2n-1}]$, qui code une matrice carrée de Hankel h.

Sortie : Une liste Stu des polynômes formels correspondants (formule 1.21).

Variables locales : m, r, p, s : compteurs ; R_0, R_1, R_2, Q, R : polynômes à coefficients dans $\mathbb{K}$.

Début

initialisation

$Stu := [\,]$;

$r :=$ l'indice du premier coefficient non nul de S ; $m := n$; $p := 2n - r - 1$;

$R_0 := 1$; $R_1 := \alpha_r + \alpha_{r+1}X + \ldots + \alpha_{p+r}X^p = \sum_{k=r}^{p+r} \alpha_k X^{k-r}$;

boucle

tant que $r < m$ **faire**

$[Q, R_2] :=$QuoResCroiss(R_0, R_1, r, p) ;

$Stu := Stu \bullet [[Q, r]]$;

$s :=$ valuation de R_2 ;

si $R_2 = 0$, on prend $s := p + 1$ car R_2 est un développement limité à l'ordre p

$p := p - s$; $m := m - r$; $r := s - r$;

$R_0 := R_1 \mod X^{p+1}$;

$R_1 := -R_2/X^s$;

fin tant que ;

si $p >= 0$ **alors**

$[R, R_2] :=$QuoResCroiss(R_1, R_0, p, p) ;

sinon

$R := 0$;

fin si ;

sortie

$Stu := Stu \bullet [[R, m]]$

Fin.

Preuve de la correction de l'algorithme 1.2

On peut se référer dans la preuve de la proposition 1.2 et de l'algorithme 1.2, à l'Exemple 1.3.2 qui est en fait un exemple suffisamment général dans lequel les degrés des quotients successifs sont irréguliers. Le lecteur pourra constater que nous reprenons la discussion exactement comme celle d'une étape quelconque de cet exemple, en donnant des noms aux exposants.

On suppose qu'on est à une étape k ($1 \leq k$), avec deux polynômes $R_0^k(X)$ et $R_1^k(X)$ obtenus à partir de l'étape $k-1$ et une matrice de Hankel d'ordre m_k définie par une liste, dont le premier élément non nul est d'indice r_k. Pour la réduire via l'algorithme 1.1, on doit inverser une matrice Toeplitz-supérieure d'ordre $2m_k - r_k$. Pour cela (lemme 1.2.1) on doit effectuer la division suivant les puissances croissantes de $R_0^k(X)$ par $R_1^k(X)$ à l'ordre $p_k = 2m_k - r_k - 1$, ce qui donne le polynôme :

$$\underbrace{q_0^k + q_1^k X + \ldots + q_{r_k}^k X^{r_k}}_{Q_k(X)} + \underbrace{c_0^k X^{s_k} + c_1^k X^{s_k+1} + \ldots + c_{p_k-s_k}^k X^{p_k}}_{X^{s_k} T_k(X)}$$

avec $c_0^k \neq 0$.

On remarque que si on arrête la division au moment où on a calculé $Q_k(X)$, on obtient :

$$\begin{array}{l|l} R_0^k(X) & R_1^k(X) \\ \cdots\cdots\cdots\cdots & \underbrace{q_0^k + q_1^k X + \ldots + q_{r_k}^k X^{r_k}}_{Q_k(X)} \\ R_2^k(X) = X^{s_k} R_k(X) & \end{array}$$

ce qui correspond à une égalité

$$R_0^k(X) = R_1^k(X) Q_k(X) + X^{s_k} R_k(X) \qquad \mod X^{p_k+1}$$

avec $\deg R_k = p_k - s_k = p_{k+1}$. Ceci implique que $T_k(X)$ peut être calculé en effectuant la division suivant les puissances croissantes de $R_k(X)$ par $R_1^k(X)_{p_{k+1}}$ à l'ordre p_{k+1}. Autrement dit

$$T_k(X) = \frac{R_k(X)}{R_1^k(X)_{p_{k+1}}} \qquad \mod X^{p_{k+1}+1}$$

Mais à l'étape suivante $(k+1)$, on doit calculer : $-\dfrac{1}{T_k(X)} \mod X^{p_{k+1}+1}$.
Il est donc inutile de calculer T_k pour l'inverser ensuite modulo $X^{p_{k+1}+1}$; en effet

$$-\frac{1}{T_k(X)} = -\frac{R_1^k(X)_{p_{k+1}}}{R_k(X)} \qquad \mod X^{p_{k+1}+1}.$$

Nous pouvons résumer cette discussion en écrivant la formule :

$$\frac{R_0^k(X)}{R_1^k(X)} = Q_k(X) + X^{s_k} \frac{R_k(X)}{R_1^k(X)_{p_{k+1}}} Q_k(X) + \frac{X^{s_k}}{\dfrac{R_1^k(X)_{p_{k+1}}}{R_k(X)}} \qquad \mod X^{p_k+1}$$

dans laquelle la division $\dfrac{R_1^k(X)_{p_{k+1}}}{R_k(X)}$ doit être effectuée seulement modulo $X^{p_{k+1}+1}$. Donc à l'étape $k+1$ on a à faire les affectations suivantes afin de réduire une matrice de Hankel d'ordre

$$m_{k+1} := m_k - r_k$$

définie par une liste, dont le premier élément non nul est d'indice

$$r_{k+1} := s_k - r_k$$

Pour cela on doit effectuer la division suivant les puissances croissantes à l'ordre

$$p_{k+1} := p_k - s_k$$

de

$$R_0^{k+1}(X) := R_1^k(X)_{p_{k+1}} = R_1^k(X) \qquad \mod X^{p_{k+1}+1}$$

par

$$R_1^{k+1}(X) := -R^k(X) = -\frac{R_2^k(X)}{X^{s_k}}$$

Complexité de l'algorithme 1.2

Le nombre d'opération arithmétiques dans la réduction d'une matrice de Hankel d'ordre n, via l'algorithme 1.2, est maximum lorsque les quotients successifs sont tous de degré 1. Cela donne le calcul de majoration suivant :

- A l'étape 1 on a à faire 1 division, $4(n-1)$ multiplications et $2n-3$ soustractions.
- A l'étape $k\,(2 \leq k \leq n-1)$ on a à faire 1 division, $4(n-k)+1$ multiplications et $4(n-k)-3$ soustractions.
- A l'étape n on a à faire seulement 1 division.

Ce qui donne n divisions, $2n^2-n-2$ multiplications et $2n^2-7n+7$ soustractions.

Proposition 1.3. *Le nombre d'opérations arithmétiques, lors de l'exécution de l'algorithme 1.2, est majoré par* $4n^2-7n+5$.

Étant donné que certains logiciels de calcul formel n'ont pas prévu de primitive pour la division des polynômes par puissances croissantes, et que dans le cas d'une matrice de Hankel associée à une suite récurrente linéaire (chapitre 3) la comparaison entre l'algorithme 1.2 et l'algorithme de Berlekamp-Massey n'est plus immédiate, on est amené à présenter une version (légèrement moins performante) de l'algorithme 1.2 dans lequel les divisions en puissances croissantes ont été remplacées par des divisions en puissances décroissantes sur des polynômes réciproques.

1.4 Une variante de l'algorithme 1.2

Il s'agit d'une constatation simple qui relie les calculs de l'algorithme 1.2 au développement en fraction continue d'une fraction rationnelle.

Par exemple si on a un développement en fraction continue usuel

$$\frac{N(X)}{D(X)} = 3X^2+1+\cfrac{1}{X^3+2X^2+\cfrac{1}{X+7}},$$

alors N et D sont premiers entre eux, $\deg(N)=2+3+1=6$, $\deg(D)=3+1=4$, $N(X)=3X^6+27X^5+43X^4+9X^3+17X^2+X+8$ et $D(X)=X^4+9X^3+14X^2+1$ et si on pose $Y=1/X$ on obtient :

$$\begin{aligned}\frac{\widehat{N}(X)}{\widehat{D}(X)} = \frac{X^6N(Y)}{X^4D(Y)}X^2\frac{N(Y)}{D(Y)} &= 3+X^2+\cfrac{X^2}{Y^3+2Y^2+\cfrac{1}{Y+7}}\\ &= 3+X^2+\cfrac{X^5}{1+2X+\cfrac{X^3}{Y+7}}\\ &= 3+X^2+\cfrac{X^5}{1+2X+\cfrac{X^4}{1+7X}}\end{aligned}$$

ce qui donne une formule analogue à la formule 1.21 (ici il faut prendre $(Q_1,d_1)=(3+X^2,2)$, $(Q_2,d_2)=(-(1+2X),3)$ et $(Q_3,d_3)=(1+7X,1)$. Plus généralement, on a le lemme utile suivant.

Lemme 1.4.1. *Soit la fraction rationnelle irréductible* $\dfrac{N(X)}{D(X)}$*, qui admet la décomposition en fraction continue suivante :*

$$\frac{N(X)}{D(X)} = q_1(X) + \cfrac{1}{q_2(X) + \cfrac{1}{\ddots\, q_{j-1}(X) + \cfrac{1}{q_j(X)}}}$$

Alors en notant $d_k = \deg q_k$ *et* $\widehat{q}_k(X) = X^{d_k} q_k(1/X)$*, on aura :*

$$\frac{\widehat{N}(X)}{\widehat{D}(X)} = \widehat{q}_1(X) + \cfrac{X^{d_1+d_2}}{\widehat{q}_2(X) + \cfrac{X^{d_2+d_3}}{\ddots\, \widehat{q}_{j-1}(X) + \cfrac{X^{d_{j-1}+d_j}}{\widehat{q}_j(X)}}} \tag{1.22}$$

Démonstration. On a : $\dfrac{N(X)}{D(X)} = q_1(X) + \cfrac{1}{q_2(X) + \cfrac{1}{\ddots\, q_{j-1}(X) + \cfrac{1}{q_j(X)}}}$

donc $\deg N = \sum_{k=1}^{j} \deg q_k$ et $\deg D = \sum_{k=2}^{j} \deg q_k$. Par passage aux polynômes réciproques et en notant $Y = 1/X$ on a :

$$\begin{aligned}\frac{\widehat{N}(Y)}{\widehat{D}(Y)} = X^{d_1}\frac{N(Y)}{D(Y)} &= X^{d_1} q_1(Y) + \cfrac{X^{d_1}}{q_2(Y) + \cfrac{1}{q_3(Y) + \cfrac{1}{\ddots\, q_{j-1}(Y) + \cfrac{1}{q_j(Y)}}}} \\ &= \widehat{q}_1(X) + \cfrac{X^{d_1+d_2}}{\widehat{q}_2(X) + \cfrac{X^{d_2}}{q_3(Y) + \cfrac{1}{\ddots\, q_{j-1}(Y) + \cfrac{1}{q_j(Y)}}}}\end{aligned}$$

et de proche en proche jusqu'à obtenir la Formule 1.22. □

Nous reprenons tout d'abord le deuxième exemple (exemple 1.3.2) donné dans la section 1.3 pour voir comment l'algorithme 1.2 page 26 doit être réécrit en l'algorithme 1.3 page 32, dans lequel on utilise les fonctions Tronk et Tronq définies comme suit.

Définition 1.4.2. *Soit* P *un polynôme de* $\mathbb{K}[X]$ *de degré* n *:* $P(X) = p_n X^n + p_{n-1}X^{n-1} + ... + p_0$*. Pour tout* $m \in \mathbb{N}$ *on définit :*

- $\mathrm{Tronk}(P, X, m)$ *: le quotient de la division (euclidienne) de* $P(X)$ *moins les monômes de degrés* $\leq m-1$ *par* X^m*.*

$$\mathrm{Tronk}(P, X, m) = \begin{cases} p_n X^{n-m} + p_{n-m-1}X^{n-m-1} + ... + p_m & \text{si } m \leq n \\ 0 & \text{sinon} \end{cases}$$

- $\mathrm{Tronq}(P, X, m)$ *: le quotient de la division (euclidienne) de* $P(X)$ *moins les monômes de degrés* $\leq m$ *par* X*.*

$$\mathrm{Tronq}(P, X, m) = \begin{cases} p_n X^{n-1} + p_{n-2}X^{n-m-1} + ... + p_{m+1}X^m & \text{si } m < n \\ 0 & \text{sinon} \end{cases}$$

on a alors : $\mathrm{Tronq}(P,X,m) = X^m\,\mathrm{Tronk}(P,X,m+1)$.

Par exemple, si $P = 5X^5+6X^4+4X^3+8X^2+9X+7$, alors

$\mathrm{Tronk}(P,X,0) = P$	$\mathrm{Tronq}(P,X,0) = 5X^4+6X^3+4X^2+8X+9$
$\mathrm{Tronk}(P,X,1) = 5X^4+6X^3+4X^2+8X+9$	$\mathrm{Tronq}(P,X,1) = 5X^4+6X^3+4X^2+8X$
$\mathrm{Tronk}(P,X,2) = 5X^3+6X^2+4X+8$	$\mathrm{Tronq}(P,X,2) = 5X^4+6X^3+4X^2$

1.4.1 Exemple

Exemple 1.3.2bis (Cet exemple reprend l'exemple 1.3.2)

- Première étape

On effectue la division suivant les puissances décroissantes de X^{17} par

$$\widehat{S}(X) = -50X^{15}-26X^{13}+34X^{11}+8X^9+X^8+20X^7-9X^6+40X^5+19X^4+8X^3-7X^2+aX+b$$

ce qui donne $X^{17} = \widehat{S}(X)\ \widehat{Q}_1(X)\ +\ Rest_1(X)$, avec :

$$\underbrace{\overbrace{10X^{13}-17X^{11}-2X^{10}+37X^9+15X^8-39X^7-11X^6-35X^5-43X^4-(24+2a)X^3+(21-2b)X^2}^{X^2\widehat{R}_1(X)}-3aX-3b}_{Rest_1(X)}$$

(le calcul des deux derniers coefficients de $Rest_1(X)$ est en fait inutile).
Avec les notations précédentes on a : $\widehat{R_1}(X) = \mathrm{Tronq}(\mathrm{Tronk}(Rest_1,X,1),X,0)$, ce qui correspond bien à 12 coefficients nécessaires pour le calcul de l'inverse de la Toeplitz-supérieure d'ordre 7 de l'étape suivante.

- Deuxième étape

On effectue la division suivant les puissances décroissantes de

$$\mathrm{Tronq}(\mathrm{Tronk}(\widehat{S},X,1),X,2) = -50X^{13}-26X^{11}+34X^9+8X^7+X^6+20X^5-9X^4+40X^3+19X^2,$$

par $-\widehat{R_1}(X)$, ce qui donne le quotient $\widehat{Q}_2(X)$ et le reste :

$$\underbrace{\overbrace{-10X^{10}-28X^8-49X^7-39X^6+8X^5-33X^4-(14+10a)X^3-(20+10b)X^2}^{X^2\widehat{R}_2(X)}-(24+2a)X+21-2b}_{Rest_2(X)}$$

On a ici

$$\widehat{R_2}(X) = \mathrm{Tronq}(\mathrm{Tronk}(Rest_2,X,1),X,0),$$

ce qui correspond bien à 9 coefficients nécessaires pour le calcul de l'inverse de la Toeplitz-supérieure d'ordre 5 de l'étape suivante.

- Troisième étape

On effectue la division suivant les puissances décroissantes de

$$\mathrm{Tronq}(\mathrm{Tronk}(-\widehat{R_1},X,1),X,1) - 10X^9+17X^7+2X^6-37X^5-15X^4+39X^3+11X^2+35X$$

par $-\widehat{R_2}(X)$, ce qui donne le quotient $\widehat{Q}_3(X) = -X$ et le reste :

$$\underbrace{45X^7-50X^6+2X^5-23X^4-29X^3+(25+10a)X^2+(10b-46)X}_{Rest_3(X)\,=\,X\,\widehat{R_3}(X)}$$

On a ici

$$\widehat{R_3}(X) = \mathrm{Tronq}(\mathrm{Tronk}(Rest_3,X,0),X,0) = \mathrm{Tronq}(Rest_3,X,0),$$

ce qui correspond bien à 7 coefficients nécessaires pour le calcul de l'inverse de la Toeplitz-supérieure d'ordre 4 de l'étape suivante.

- Quatrième étape

On effectue la division suivant les puissances décroissantes de

$$\text{Tronq}(\text{Tronk}(-\widehat{R_2}, X, 0), X, 1) = \text{Tronq}(-\widehat{R_2}, X, 1) = 10X^7+28X^5+49X^4+39X^3-8X^2+33X$$

par $-\widehat{R_3}(X)$, ce qui donne le quotient $\widehat{Q}_4(X)$ et le reste :

$$\underbrace{\overbrace{(36+9a)X^2-(44-10a-9b)X}^{X\widehat{R_4}(X)}\ -46+10b}_{Rest_4(X)}$$

On a ici

$$\widehat{R_4}(X) = \text{Tronq}(\text{Tronk}(Rest_4, X, 0), X, 0) = \text{Tronq}(Rest_4, X, 0),$$

ce qui correspond bien à un polynôme de degré 1 nécessaire pour la dernière étape.

- Dernière étape

Les derniers coefficients de notre réduite sont obtenus en effectuant la division de $-X\widehat{R_4}(X) = -(36+9a)X^2+(44-10a-9b)X$ par $\text{Tronk}(\text{Tronq}(\text{Tronk}(-\widehat{R_3}, X, 0), X, 4), X, 4) = -45X+50$ qui donne un quotient égal à $(21-20a)X-(36+22a+20b) = XR(\frac{1}{X})$.

Ainsi on obtient l'algorithme 1.3 page suivante.

1.4.2 Algorithme 1.3

La sortie de l'algorithme 1.3 est une suite de quotients, notée $Stu = [\widehat{Q}_1, \ldots, \widehat{Q}_k, (X^t R(\frac{1}{X}), d_R)]$, dans un algorithme d'Euclide « tronqué ». On notera que :

- les $\widehat{Q}_i(X)$ dans l'algorithme 1.3 sont les polynômes réciproques de ceux calculés par l'algorithme 1.2.
- dans l'algorithme 1.2 il était nécessaire que les quotients soient des polynômes formels tandis que dans l'algorithme 1.3 les polynômes réciproques sont des polynômes usuels (cf. la formule 1.22).

Dans cette variante, si $\widehat{Q}_i(X) = c_{0,i}X^{d_i} + \ldots + c_{d_i,i}$, alors le $i^{\text{ème}}$ bloc D_{ii} de D est d'ordre $\deg \widehat{Q}_i(X)$ et est donné par :

$$D_{ii} = \text{Hki}(c_{0,i}, \ldots, c_{d_i-1,i}).$$

Et en ce qui concerne le dernier bloc $D_{k+1,k+1}$, si $X^t R(\frac{1}{X}) = r_0X^t + \ldots + r_t$ alors :

$$D_{k+1,k+1} = \text{Hki}(\underbrace{0, \ldots, 0}_{d_R-t-1}, r_0, \ldots, r_t).$$

La correction de l'algorithme 1.3 découle immédiatement de celle de l'algorithme 1.2.

Complexité de l'algorithme 1.3

Le nombre d'opération arithmétiques dans la réduction d'une matrice de Hankel d'ordre n, via l'algorithme 1.3, est maximum lorsque les quotients successifs sont tous de degré 1. Cela donne le calcul de majoration suivant :

- A l'étape 1 on a à faire 1 division, $4(n-1)$ multiplications et $2n-3$ soustractions.

Algorithme 1.3. ***Algorithme de réduction d'une matrice de Hankel (3)***

Entrée : Une liste non nulle d'éléments du corps $\mathbb{K}$, $S = [\alpha_1, \alpha_2, \ldots, \alpha_{2n-1}]$, qui code une matrice carrée de Hankel h.

Sortie : Une liste Stu de polynômes, qui codent les blocs diagonaux « Hankel-inférieurs ».

Variables locales : $m, r, d_0, d_1, d_2, dd_1, dd_2$: compteurs ; R_0, R_1, R_2, Q, R : polynômes.

Début

initialisation

$m := n$; $R_0 := X^{2n-1}$; $R_1 := \alpha_1 X^{2n-2} + \alpha_2 X^{2n-3} + \ldots + \alpha_{2n-1} = \sum_{k=0}^{2n-2} \alpha_{k+1} X^{2n-2-k}$; $Stu := [\,]$;

$d_0 := \deg R_0$; $d_1 := \deg R_1$;

boucle

tant que $2d_1 - d_0 > 0$ **faire**

$dd_1 := d_0 - d_1 - 1$;

$Q :=$ le quotient de la division de R_0 par R_1 ;

$R_2 :=$ le reste de la division de R_0 par R_1 ;

$R_1 := \mathrm{Tronk}(R_1, X, dd_1)$;

pour Tronk et Tronq voir Définition 1.4.2 page 29.

$R_2 := \mathrm{Tronk}(-R_2, X, dd_1)$;

$d_1 := \deg R_1$; $d_2 := \deg R_2$; $dd_2 := d_1 - d_2$;

$R_0 := \mathrm{Tronq}(R_1, X, dd_2)$;

$R_1 := \mathrm{Tronq}(R_2, X, 0)$;

$Stu := Stu \bullet [Q]$;

$r := \deg Q$; $m := m - r$;

$d_0 := \deg R_0$; $d_1 := \deg R_1$;

fin tant que

si $d_1 >= 0$ **alors**

$R_1 := X^{d_1} R_1$; $R_0 := \mathrm{Tronk}(R_0, X, d_0 - d_1)$; $R :=$ le quotient de la division de R_1 par R_0 ;

sinon

$R := 0$;

fin si ;

sortie

Return $Stu := Stu \bullet [[R, m]]$;

Fin.

- A l'étape $k\,(2 \leq k \leq n-1)$ on a à faire 1 division, $4(n-k)+1$ multiplications et $4(n-k)-1$ soustractions.
- A l'étape n on a à faire seulement 1 division.

Ce qui donne n divisions, $2n^2 - n - 2$ multiplications et $2n^2 - 5n + 3$ soustractions.

Proposition 1.4. *Le nombre d'opérations arithmétiques, lors de l'exécution de l'algorithme 1.3, est majoré par* $4n^2 - 5n + 1$.

Le léger excédent par rapport à l'algorithme 1.2 est dû au fait qu'on a calculé des coefficients constants inutiles dans les restes.

1.5 Comparaison avec l'algorithme classique

Dans [9] et [22], on peut trouver le résultat classique de diagonalisation par blocs d'une matrice de Hankel (proposition 1.1 Equation (1.1)) et un algorithme qui a comme outil le développement en fraction continue de $-1/\sum \alpha_i x^{-i}$.

Par ailleurs, il est utile de clarifier la différence entre la diagonalisation par blocs obtenue avec l'algorithme classique et celle obtenue par l'algorithme 1.2.

Etant donné une matrice de Hankel $h = \mathrm{Hk}(\alpha_1, \ldots, \alpha_{2n-1})$. Nous désignons par $D_{\text{class}} = {}^{\mathrm{t}}R_{\text{class}} h R_{\text{class}}$ (resp. $D = {}^{\mathrm{t}}AhA$) la réduite diagonale par blocs de h via l'algorithme classique (resp. l'algorithme 1.2).

Pour la comparaison nous nous reportons à l'exemple qui vient.

Exemple 1.5.1.
Soit h la matrice de Hankel d'ordre 15 définie par la liste :

$$S = [34, -12, 28, 2, -16, 41, 18, -41, -40, -23, 32, 6, -15, 2, -22, -21, -45, 11, 50, -15, -32, -5, 30, 18, -9, 40, 35, 1, -31],$$

Les mineurs principaux dominants de h (modulo 101) sont

$$\{34, 0, 0, -26, 21, 0, -24, 0, 0, -49, -18, 0, 0, 0, -29\},$$

par conséquent les blocs de la réduite seront d'ordre respectif : $1, 3, 1, 2, 3, 1, 4$.

- Avec l'algorithme 1.2 on a :

la matrice de passage,

$$A = \begin{pmatrix}
3 & -7 & 21 & 14 & 46 & 20 & 46 & -48 & 0 & 0 & -43 & 12 & 0 & 0 & 0 \\
0 & -3 & 2 & 27 & -38 & 24 & -46 & 30 & -48 & 0 & -48 & 0 & 12 & 0 & 0 \\
0 & 0 & -3 & 2 & -47 & 34 & -19 & 5 & 30 & -48 & 28 & -24 & 0 & 12 & 0 \\
0 & 0 & 0 & -3 & 4 & 14 & 46 & -33 & 5 & 30 & 40 & 11 & -24 & 0 & 12 \\
0 & 0 & 0 & 0 & -6 & 5 & -25 & 5 & -33 & 5 & -19 & -37 & 11 & -24 & 0 \\
0 & 0 & 0 & 0 & 0 & 35 & 35 & -23 & 5 & -33 & 47 & 12 & -37 & 11 & -24 \\
0 & 0 & 0 & 0 & 0 & 0 & 35 & 39 & -23 & 5 & -44 & 12 & 12 & -37 & 11 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 39 & 39 & -23 & 20 & 22 & 12 & 12 & -37 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 39 & 39 & 41 & 20 & 22 & 12 & 12 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 39 & -30 & 41 & 20 & 22 & 12 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & 41 & 20 & 22 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & 41 & 20 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & 41 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30 & -30 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30
\end{pmatrix},$$

et la réduite diagonale par blocs,

$$D = \begin{pmatrix} (3) & & & & & & & \\ & \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 3 \\ -1 & 3 & 2 \end{pmatrix} & & & & & & \\ & & (2) & & & & & \\ & & & \begin{pmatrix} 0 & 11 \\ 11 & -5 \end{pmatrix} & & & & \\ & & & & \begin{pmatrix} 0 & 0 & 4 \\ 0 & 4 & 0 \\ 4 & 0 & 0 \end{pmatrix} & & & \\ & & & & & (7) & & \\ & & & & & & \begin{pmatrix} 0 & 0 & 0 & 20 \\ 0 & 0 & 20 & 8 \\ 0 & 20 & 8 & -17 \\ 20 & 8 & -17 & -15 \end{pmatrix} \end{pmatrix}.$$

- Avec l'algorithme classique on a :

la matrice de passage classique,

$$R_{\text{class}} = \begin{pmatrix} 1 & 36 & 0 & 0 & 26 & 15 & 0 & -9 & 0 & 0 & 25 & 40 & 0 & 0 & 0 \\ 0 & 1 & 36 & 0 & 40 & 18 & 15 & -7 & -9 & 0 & 42 & 0 & 40 & 0 & 0 \\ 0 & 0 & 1 & 36 & -9 & -25 & 18 & -18 & -7 & -9 & 26 & 21 & 0 & 40 & 0 \\ 0 & 0 & 0 & 1 & 33 & -40 & -25 & 38 & -18 & -7 & -35 & 3 & 21 & 0 & 40 \\ 0 & 0 & 0 & 0 & 1 & 29 & -40 & -18 & 38 & -18 & 4 & 45 & 3 & 21 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 29 & 2 & -18 & 38 & 22 & 40 & 45 & 3 & 21 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & -18 & -12 & 40 & 40 & 45 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 & 40 & 45 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 & 40 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

ou encore avec les notations de la proposition 1.1

$$R_{\text{class}} = (\underbrace{\mathbf{q}_0}_{Bloc1}\ ,\ \underbrace{\mathbf{q}_1, J_{15}\mathbf{q}_1, J_{15}^2\mathbf{q}_1}_{Bloc2}\ ,\ \underbrace{\mathbf{q_2}}_{Bloc3}\ ,\ \underbrace{\mathbf{q}_3, J_{15}\mathbf{q}_3}_{Bloc4}\ ,\ \underbrace{\mathbf{q}_4, J_{15}\mathbf{q}_4, J_{15}^2\mathbf{q}_4}_{Bloc5}\ ,\ \underbrace{\mathbf{q}_5}_{Bloc6}\ ,\ \underbrace{\mathbf{q}_6, J_{15}\mathbf{q}_6, J_{15}^2\mathbf{q}_6, J_{15}^3\mathbf{q}_6}_{Bloc7}\ ),$$

et la réduite diagonale par blocs,

$$D_{\text{class}} = \begin{pmatrix} (34) & & & & & & \\ & \begin{pmatrix} 0 & 0 & -45 \\ 0 & -45 & -34 \\ -45 & -34 & 10 \end{pmatrix} & & & & & \\ & & (-28) & & & & \\ & & & \begin{pmatrix} 0 & -38 \\ -38 & 47 \end{pmatrix} & & & \\ & & & & \begin{pmatrix} 0 & 0 & -33 \\ 0 & -33 & 0 \\ -33 & 0 & 0 \end{pmatrix} & & \\ & & & & & (-12) & \\ & & & & & & \begin{pmatrix} 0 & 0 & 0 & 9 \\ 0 & 0 & 9 & 44 \\ 0 & 9 & 44 & -43 \\ 9 & 44 & -43 & -32 \end{pmatrix} \end{pmatrix}.$$

(h est bien LU-équivalente à D_{class}). Pour faire la comparaison entre les deux méthodes, on doit rendre $A = (a_{i,j})$ unitriangulaire supérieure, d'où une nouvelle matrice de passage A_{un} définie par :

$$A_{\text{un}} \ = \ A \ \times \ \text{Diag}\left(\frac{1}{a_{1,1}}, \frac{1}{a_{2,2}}, \ldots, \frac{1}{a_{15,15}}\right).$$

Ce qui donne pour notre exemple :

$$A_{\text{un}} = \begin{pmatrix}
1 & 36 & -7 & 29 & 26 & 15 & -16 & -9 & 0 & 0 & 25 & 40 & 0 & 0 & 0 \\
0 & 1 & 33 & -9 & 40 & 18 & 16 & -7 & -9 & 0 & 42 & 0 & 40 & 0 & 0 \\
0 & 0 & 1 & 33 & -9 & -25 & 11 & -18 & -7 & -9 & 26 & 21 & 0 & 40 & 0 \\
0 & 0 & 0 & 1 & 33 & -40 & -16 & 38 & -18 & -7 & -35 & 3 & 21 & 0 & 40 \\
0 & 0 & 0 & 0 & 1 & 29 & -44 & -18 & 38 & -18 & 4 & 45 & 3 & 21 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & -18 & 38 & 22 & 40 & 45 & 3 & 21 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & -18 & -12 & 40 & 40 & 45 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 & 40 & 45 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 & 40 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix},$$

ou encore,

$$A_{\text{un}} = (\underbrace{\mathbf{col}_0}_{Bloc1}\ ,\ \underbrace{\mathbf{col}_1, \mathbf{col}_2, \mathbf{col}_3}_{Bloc2}\ ,\ \underbrace{\mathbf{col}_4}_{Bloc3}\ ,\ \underbrace{\mathbf{col}_5, \mathbf{col}_6}_{Bloc4}\ ,\ \underbrace{\mathbf{col}_7, \mathbf{col}_8, \mathbf{col}_9}_{Bloc5}\ ,\ \underbrace{\mathbf{col}_{10}}_{Bloc6}\ ,\ \underbrace{\mathbf{col}_{11}, \mathbf{col}_{12}, \mathbf{col}_{13}, \mathbf{col}_{14}}_{Bloc7}\ ),$$

ce qui génère la réduite diagonale par blocs $D_{\text{un}} = {}^{\text{t}}A_{\text{un}} h A_{\text{un}}$, suivante :

$$D_{\text{un}} = \begin{pmatrix}
(34) & & & & & & \\
& \begin{pmatrix} 0 & 0 & -45 \\ 0 & -45 & 34 \\ -45 & 34 & -11 \end{pmatrix} & & & & & \\
& & (-28) & & & & \\
& & & \begin{pmatrix} 0 & -38 \\ -38 & -47 \end{pmatrix} & & & \\
& & & & \begin{pmatrix} 0 & 0 & -33 \\ 0 & -33 & 0 \\ -33 & 0 & 0 \end{pmatrix} & & \\
& & & & & (-12) & \\
& & & & & & \begin{pmatrix} 0 & 0 & 0 & 9 \\ 0 & 0 & 9 & 44 \\ 0 & 9 & 44 & -43 \\ 9 & 44 & -43 & -32 \end{pmatrix}
\end{pmatrix}.$$

(et h est aussi LU-équivalente à D_{un}).

□

On a alors les différences et les ressemblances suivantes :

- On a bien le deuxième résultat de la proposition 1.1 : $\mathbf{col}_{m_i} = \mathbf{q}_i, \quad i = 0, \ldots, 6$.
- Les matrices de passage A et A_{un} ne sont pas nécessairement Toeplitz par blocs alors que R_{class} est Toeplitz par blocs.
  Pour avoir R_{class}, il suffit donc de déterminer la première colonne de chacun de ses blocs, c'est-à-dire les $\mathbf{q}_i$ qui sont entièrement déterminer par les $\mathbf{col}_{k_i}$.
  Ainsi, la matrice de passage de l'algorithme 1.2 (rendue unitriangulaire supérieure) permet de retrouver facilement la matrice de passage de la méthode classique.
- Dans D_{class}, le premier bloc est inchangé (c'est celui de h), alors que dans D c'est l'inverse de celui de h.
- Avec l'algorithme 1.2, il n'est pas nécessaire de calculer A pour obtenir D.

Remarque 1.1. *On sait que la LU-décomposition d'une matrice carrée régulière h dont tous les mineurs principaux dominants sont non nuls, existe et est unique. Si en outre h est une matrice de Hankel cette LU-décomposition est la même chose que la décomposition obtenue ci-dessus par la méthode classique (introduite dans la section 1.1), chaque bloc diagonal Hankel inférieur étant de taille 1.*

1.6 Application : Preuve élémentaire du Théorème de Frobenius

Soit $\mathbb{K}$ un corps ordonné. Nous donnons ici, une preuve simple, plus directe que celle donnée par *Gantmacher* (voir [17] chap. 10) et purement algébrique et différente à celle donnée par *Fuhrmann* (voir [16] chap. 8) du théorème de Frobenius, qui calcule la signature d'une matrice de Hankel réelle à partir des signes de ces mineurs principaux dominants.
Dans la suite on aura besoin de la notion de signe d'un 'élément de $\mathbb{K}$ et d'une propriété de la signature d'une matrice diagonale par bloc.

Définition 1.6.1. *Soit $a \in \mathbb{K}$, le signe de l'élément a est défini par :*

$$\text{sig}(a) = \begin{cases} 0 & \text{si } a = 0 \\ +1 & \text{si } a > 0 \\ -1 & \text{si } a < 0 \end{cases}$$

Propriété
Soit M une matrice symétrique à coefficient dans $\mathbb{K}$. la signature de M noté par $\text{Sig}(M)$, vérifie :

$$\text{Sig}(M) = \begin{cases} \text{sig}(a) & \text{si } M = [a] \\ \sum_{i=1}^{n} \text{Sig}(M_i) & \text{si } M = \text{Diag}(M_1, M_2, \ldots, M_n) \end{cases}$$

Le lemme suivant montre qu'on ne change pas le signe des mineurs principaux dominants d'une matrice symétrique lorsqu'on utilise pour changement de base une matrice triangulaire supérieure.

Lemme 1.6.2. *Soient M une matrice symétrique et N une matrice régulière de même ordre que M, partagées en sous-matrices (de mêmes ordre respectifs) de la manière suivante :*

$$M = \begin{pmatrix} A & B \\ {}^tB & C \end{pmatrix}, \qquad N = \begin{pmatrix} E & F \\ 0 & G \end{pmatrix}.$$

Alors,

$${}^tNMN = \begin{pmatrix} {}^tEAE & X \\ {}^tX & Y \end{pmatrix},$$

où : $X = {}^tEAF + {}^tEBG$,
$Y = {}^tFAF + {}^tGCG + {}^tG\,{}^tBF + {}^tFBG$.

Enfin : $\operatorname{sig}(\det({}^tEAE)) = \operatorname{sig}(\det(A))$ *et* $\operatorname{Sig}({}^tEAE) = \operatorname{Sig}(A)$.

Lemme 1.6.3. *Soit h une matrice de Hankel et D sa réduite donnée par l'algorithme 1.1. Alors leurs mineurs principaux dominants respectifs sont ègaux « à la multiplication par des carrés non nuls près », en particulier les mineurs correspondants sont de mêmes signes :* $\operatorname{sig}(\Delta_k(h)) = \operatorname{sig}(\Delta_k(D))$ *pour* $k = 1, \ldots, n$.

Dans cette section nous considérons une matrice carrée d'ordre n, *régulière* et de Hankel h. Sa réduite diagonale par blocs Hankel-inférieurs, D, codée par la liste L que calcule l'algorithme 1.1 est donc de la forme :

$$D = \operatorname{Diag}(D_{11}, D_{22}, \ldots, D_{kk})$$

où k est la longueur de L et $D_{ii} = \operatorname{Hki}(\mathrm{L}[i])$ pour $1 \leq i \leq k$. Désignons par r_i la longueur de $\mathrm{L}[i]$ et notons pour $i = 1, \ldots, k$, $m_i = \sum_{j=1}^{i}(r_j)$, en particulier $m_0 = 0$. Puisque D est diagonale par blocs, le signe du déterminant de chaque bloc D_{ii} est donné par :

$$\operatorname{sig}(\det(D_{ii})) = \operatorname{sig}(\Delta_{m_i}(D)\,\Delta_{m_{i-1}}(D)).$$

Pour achever l'idée de la démonstration du théorème de Frobenius, nous avons besoin d'un calcul qui donne la signature d'une matrice Hankel-inférieure. Pour cela nous nous référons à l'exemple suivant.

Exemple 1.6.4. Considérons la matrice Hankel-inférieure,

$$H_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 & a & b \\ 0 & 0 & 1 & a & b & c \\ 0 & 1 & a & b & c & d \\ 1 & a & b & c & d & e \end{pmatrix}.$$

Posons,

$$P_1 = \begin{pmatrix} 1 & -a & -b & -c & -d & -e/2 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Si nous utilisons P_1 comme matrice de changement de base pour H_1 on aura :

$$ {}^tP_1\, H_1\, P_1 = \begin{pmatrix} 0&0&0&0&0&1\\ 0&0&0&0&1&0\\ 0&0&0&1&a&0\\ 0&0&1&a&b&0\\ 0&1&a&b&c&0\\ 1&0&0&0&0&0 \end{pmatrix}. $$

Par conséquent, si nous posons

$$ P = P_1 \begin{pmatrix} 1&0&0&0&0&0\\ 0&1&-a&-b&-c/2&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&-a/2&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{pmatrix}, $$

on aura

$$ {}^tP\, H_1\, P = \begin{pmatrix} 0&0&0&0&0&1\\ 0&0&0&0&1&0\\ 0&0&0&1&0&0\\ 0&0&1&0&0&0\\ 0&1&0&0&0&0\\ 1&0&0&0&0&0 \end{pmatrix}, $$

qui est trivialement diagonalisable en la matrice $\mathrm{Diag}(1,1,1,-1,-1,-1)$.

Remarque 1.2. *Signalons que la méthode décrite dans l'exemple 1.6.4 est vraie pour tout ordre.*

Ainsi, le lemme suivant nous semble être immédiat.

Lemme 1.6.5. *Soient $\mathbb{K}$ un corps ordonné, $(a_1, a_2, \ldots, a_n) \in \mathbb{K}^n$ avec $a_1 \neq 0$ et $M = \mathrm{Hki}(a_1, a_2, \ldots, a_n)$. On a alors :*

$$ \mathrm{Sig}(M) = \begin{cases} 0 & \text{si } n \text{ est pair} \\ \mathrm{sig}(a_1) = (-1)^{\frac{n-1}{2}}\, \mathrm{sig}(\det(M)) & \text{si } n \text{ est impair} \end{cases} $$

Toutes ces observations, le lemme 1.6.3 et le lemme 1.6.5 donnent donc le théorème suivant :

Théorème 1.3. (Frobenius)
Soit h une matrice de Hankel régulière, d'ordre n. Soit $m_0 = 0$ et $1 \leq m_1 < \cdots < m_k = n$ les ordres des mineurs principaux dominants non nuls de h. Posons $r_i = m_i - m_{i-1}$ pour $i = 1, \ldots, k$.
Alors la signature de h est donnée par la formule suivante :

$$ \mathrm{Sig}(h) = \sum_{i=1}^{k} \begin{cases} 0, & \text{si } r_i \text{ est pair} \\ (-1)^{\frac{r_i-1}{2}}\, \mathrm{sig}(\Delta_{m_i}(h)\, \Delta_{m_{i-1}}(h)), & \text{si } r_i \text{ est impair} \end{cases} $$

2. Algorithme d'Euclide signé et diagonalisation par blocs des formes de Hankel et de Bezout

Introduction

Ce chapitre est consacré à l'analyse de la « diagonalisation par blocs » des matrices de Hankel et de Bezout associées à deux polynômes.

Plus précisement, soit $U(X)$ et $V(X)$ deux polynômes sur $\mathbb{K}[X]$ telle que $n = \deg(U(X)) > \deg(V(X))$. Soit $\mathrm{H}(U,V)$ (respectivement $\mathrm{Bez}(U,V)$) la matrice de Hankel (respectivement de Bezout) associée à $U(X)$ et $V(X)$. Dans [8] et [7], les auteurs introduisent un algorithme parallèle pour la diagonalisation par blocs de $\mathrm{JBez}(U,V)\mathrm{J}$ et expliquent comment obtenir la suite des restes apparue dans l'Algorithme d'Euclide signé appliqué à $U(X)$ et $V(X)$ durant l'exécution de l'algorithme.

En outre, comme

$$\mathrm{Bez}(U,V) \;=\; \mathrm{Bez}(U,1)\,\mathrm{H}(U,V)\,\mathrm{Bez}(U,1) \tag{2.1}$$

(pour la preuve, voir [26], page 475) la diagonalisation par blocs de $\mathrm{JBez}(U,V)\mathrm{J}$ peut être obtenue à partir de la diagonalisation par blocs de $\mathrm{H}(U,V)$ et vice versa.

Ici nous montrons que l'application de notre algorithme (introduit dans le Chap 1) à $\mathrm{H}(U,V)$ rapporte une preuve plus simple et plus belle du parallélisme entre l'Algorithme d'Euclide signé appliqué à $U(X)$ et $V(X)$ et la diagonalisation par blocs de la matrice de Hankel ssociée $\mathrm{H}(U,V)$. De plus, nous devons comparer nos résultats et la version séquentielle de l'algorithme présenté dans [7] pour des matrices de Bezout. Nous montrons que les deux algorithmes donnent la même matrice diagonale par blocs, ce qui est différent de la méthode classique.

Après un rappel (section 2.1) concernant les matrices de Hankel et de Bezout associées à deux polynômes nous présentons (section 2.2 la diagonalisation par blocs de $\mathrm{H}(U,V)$. Nous analysons ensuite la diagonalisation par blocs des matrices de Bezout.

2.1 Matrices de Hankel et de Bezout associées à deux polynômes

Dans $\mathbb{K}[X]$ nous considérons deux polynômes :

$$U(X) = u_n X^n + u_{n-1}X^{n-1} + \ldots + u_1 X + u_0 \text{ et } V(X) = v_m X^m + v_{m-1}X^{m-1} + \ldots + v_1 X + v_0,$$

de degrè respectif n et m telle que : $n > m > 0$.

Rappelons que l'Algorithme d'Euclide signé, appliqué à $U(X)$ et $V(X)$ génère une suite des quotients $\{q_i(X)\}$ et des restes $\{r_i(X)\}$ de la manière suivante :

$$\begin{cases} r_{-1}(X) := U(X), \quad r_0(X) := V(X) \\ r_{i-2}(X) := r_{i-1}(X)q_i(X) - r_i(X); \ \ \deg(r_i(X)) < \deg(r_{i-1}(X)), \ 1 \leq i \leq L \end{cases} \tag{2.2}$$

où $-r_i(X)$ est le reste de la division de $r_{i-2}(X)$ par r_{i-1} et $r_L = \text{pgcd}(U(X), V(X))$.

Dans cette section, on aura besoin (en plus de la *base standard* $\mathcal{B}_{St} = \{1, \ldots, X^{n-1}\}$) de la notion de *base de Horner* de $\mathbb{K}_{n-1}[X]$, définie par les polynômes de Horner.

Définition 2.1.1. *On appelle polynômes de Horner associés à un polynôme* $U(X) = u_nX^n + u_{n-1}X^{n-1} + \ldots + u_1X + u_0$ *de degré* n, *les polynômes* $\{\text{Hor}_1(X), \text{Hor}_2(X), \ldots, \text{Hor}_n(X)\}$, *définis par :*

$$\text{Hor}_i(X) = u_nX^{n-i} + u_{n-1}X^{n-i-1} + \ldots + u_{i+1}X + u_i, \qquad i = 1, \ldots, n.$$

D'une manière récursive on a :

$$\begin{aligned} &\text{Hor}_n(X) = 1, \\ &\text{Hor}_{n-k}(X) = X\text{Hor}_{n-k+1}(X) + u_{n-k}, \qquad k = 1, \ldots, n-1. \end{aligned}$$

Remarque 2.1. *Remarquons que la famille* $\{\text{Hor}_1(X), \text{Hor}_2(X), \ldots, \text{Hor}_n(X)\}$ *est libre et maximale dans* $\mathbb{K}_{n-1}[X]$, *elle forme donc une base : la* base de Horner *(appelée également* base de controle *dans [16] page* 98*) de* $\mathbb{K}_{n-1}[X]$ *associée au polynôme* $U(X)$, *notée* $\mathcal{B}_{Ho}$. *Ainsi la matrice de changement de base de* $\mathcal{B}_{Ho}$ *à* $\mathcal{B}_{St}$ *est :*

$$H_S = \begin{pmatrix} u_1 & u_2 & \cdots & u_{n-1} & u_n \\ u_2 & u_3 & \cdots & u_n & 0 \\ \vdots & \vdots & \iddots & \iddots & \vdots \\ u_{n-1} & u_n & \iddots & & \vdots \\ u_n & 0 & \cdots & \cdots & 0 \end{pmatrix} = \text{Hks}(u_1, \cdots, u_{n-1}, u_n).$$

Soit φ_U l'application linéaire correspondante à la multiplication par X modulo $U(X)$ sur $\mathbb{K}_{n-1}[X]$:

$$\begin{aligned} \varphi_U : \mathbb{K}_{n-1}[X] &\longrightarrow \mathbb{K}_{n-1}[X] \\ P(X) &\longmapsto \text{rest}(XP(X), U(X)) \end{aligned}$$

Définition 2.1.2. *On appelle matrice Compagnon de* $U(X)$, *la matrice* $Comp_U$ *de l'application linéaire* φ_U *relativement à la base standard, suivante :*

$$Comp_U = M_{(\varphi_U, \mathcal{B}_{St}, \mathcal{B}_{St})} = \begin{pmatrix} 0 & 0 \cdots 0 & -u_0/u_n \\ 1 & 0 \cdots 0 & -u_1/u_n \\ 0 & 1 \cdots 0 & -u_2/u_n \\ \vdots & \vdots \ddots \vdots & \vdots \\ 0 & 0 \cdots 1 & -u_{n-1}/u_n \end{pmatrix}.$$

Plus généralement, considérons l'application linéaire Φ_U^V correspondante à la multiplication par $V(X)$ modulo $U(X)$ sur $\mathbb{K}_{n-1}[X]$:

$$\begin{array}{rccl}\Phi_U^V : & \mathbb{K}_{n-1}[X] & \longrightarrow & \mathbb{K}_{n-1}[X] \\ & P(X) & \longmapsto & \text{rest}(V(X)P(X), U(X))\end{array}$$

Alors on a :

$$V(Comp_U) = M_{(\Phi_U^V, \mathcal{B}_{St}, \mathcal{B}_{St})}$$

Remarque 2.2.

1. $\varphi_U = \Phi_U^X$.
2. *Si* $V(X) = 1$ *alors* $\Phi_U^1 = \text{id}_{\mathbb{K}_{n-1}[X]}$ *et par conséquent :*

$$M_{(\Phi_U^1, \mathcal{B}_{Ho}, \mathcal{B}_{St})} = H_S.$$

2.1.1 La matrice H(U,V)

Nous désignons par $\mathcal{S}(X)$ la série formelle associée à la fonction $\dfrac{V(X)}{U(X)}$:

$$\mathcal{S}(X) = \frac{V(X)}{U(X)} = \sum_{i=1}^{\infty} h_i X^{-i} \in \mathbb{K}[[\frac{1}{X}]].$$

Définition 2.1.3. *On appelle matrice de Hankel associée à* $U(X)$ *et* $V(X)$*, la matrice carrée d'ordre* n*, définie par* $\text{Hk}(h_1, h_2, \cdots, h_{2n-1})$ *et notée par* $\text{H}(U,V)$*. C'est-à-dire,* $\text{H}(U,V) = (h_{i+j-1})_{1\leq i,j\leq n}$ *:*

$$\text{H}(U,V) = \begin{pmatrix} h_1 & h_2 & \cdots & h_{n-1} & h_n \\ h_2 & h_3 & \cdots & h_n & h_{n+1} \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-3} & h_{2n-2} \\ h_n & h_{n+1} & \cdots & h_{2n-2} & h_{2n-1} \end{pmatrix}.$$

Pour plus de détailles concernant ces matrices et leurs propriétés on peut consulter, par exemple, [24](1989), [8](1994), [12](2002), [3](2003) et [13](2004).
Mais comme premier résultat, on peut dire que la matrice $H(U,V)$ n'est autre que la matrice de l'application linéaire Φ_U^V de la base standard à la base de Horner, qu'on peut l'écrire par :

$$H(U,V) = M_{(\Phi_U^V, \mathcal{B}_{St}, \mathcal{B}_{Ho})}$$

Exemple 2.1.1.
Dans $\mathbb{Q}[X]$ *on considère les polynômes :*

$$U(X) = X^5 - 2X^4 + 6X^3 - X + 5, \qquad V(X) = -7X^4 + X^3 + 2X^2 - 5X - 6,$$

la matrice de Hankel, associée à $U(X)$ *et* $V(X)$ *est alors :*

$$\text{H}(U,V) = \begin{pmatrix} -7 & -13 & 18 & 109 & 97 \\ -13 & 18 & 109 & 97 & -438 \\ 18 & 109 & 97 & -438 & -1375 \\ 109 & 97 & -438 & -1375 & -103 \\ 97 & -438 & -1375 & -103 & 7596 \end{pmatrix}.$$

□

Réciproquement, toute matrice de Hankel régulière peut être considérée comme étant la matrice de Hankel associée à deux polynômes premiers entre eux. Ci-après nous présentons la version donnée dans [8].

Proposition 2.1. *Pour toute matrice de Hankel $h = \mathrm{Hk}(h_1, \ldots, h_{2n-1})$ d'ordre n, régulière, il existe deux polynômes $U(X)$ et $V(X)$ premiers entre eux, de degré respectif $n > m$, tels que :*

$$h = \mathrm{H}(U, V).$$

Les polynômes $U(X)$ et $V(X)$ sont liés à h par les équations suivantes :

$$h \begin{pmatrix} u_0 \\ \vdots \\ u_{n-1} \end{pmatrix} = -u_n \begin{pmatrix} h_{n+1} \\ \vdots \\ h_{2n} \end{pmatrix},$$

$$\begin{pmatrix} v_{n-1} \\ v_{n-2} \\ \vdots \\ v_0 \end{pmatrix} = \begin{pmatrix} h_1 & 0 & \cdots & 0 \\ h_2 & h_1 & & \vdots \\ \vdots & \vdots & \ddots & 0 \\ h_n & h_{n-1} & \cdots & h_1 \end{pmatrix} \begin{pmatrix} u_n \\ u_{n-1} \\ \vdots \\ u_1 \end{pmatrix},$$

où h_{2n} est à fixé dans $\mathbb{K}$.

Pour la preuve on peut consulter [7, 15].

Remarque 2.3. *Utilisons les notations de la Proposition 2.1 et considérons la matrice Hankel-inférieure :*

$$h = \begin{pmatrix} 0 & \cdots & 0 & h_n \\ \vdots & & h_n & h_{n+1} \\ 0 & \iddots & \vdots & \vdots \\ h_n & \cdots & h_{2n-2} & h_{2n-1} \end{pmatrix}.$$

Alors

$$\begin{pmatrix} v_{n-1} \\ v_{n-2} \\ \vdots \\ v_0 \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ h_n & \cdots & 0 \end{pmatrix} \begin{pmatrix} u_n \\ u_{n-1} \\ \vdots \\ u_1 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ h_n u_n \end{pmatrix},$$

ainsi $V(X)$ est une constante de $\mathbb{K}$. Si on prend $V(X) = v_0 = 1$ alors $u_n = 1/h_n$ et $h = \mathrm{H}(U, 1)$. Nous utiliserons cette remarque après dans la section 2.3.1.

Dans la littérature on trouve que les matrices de Hankel sont étroitement liées aux matrices de Bezout. Pour cela, on donne la définitions suivantes :

2.1.2 La matrice Bez(U,V)

Définition 2.1.4. *On appelle matrice de Bezout associée aux polynômes $U(X)$ et $V(X)$, la matrice symétrique*

$$\mathrm{Bez}(U, V) = \begin{pmatrix} c_{0,0} & \cdots & c_{0,n-1} \\ \vdots & & \vdots \\ c_{n-1,0} & \cdots & c_{n-1,n-1} \end{pmatrix},$$

où les $c_{i,j}$ sont définis par l'expression de Cayley :

$$\frac{U(X)\,V(Y)-U(Y)\,V(X)}{X-Y} = \sum_{i,j=0}^{n-1} c_{i,j}X^iY^j$$

Comme dans le cas des matrices de Hankel associées à deux polynômes, on peut dire que la matrice $\mathrm{Bez}(U,V)$ n'est autre que la matrice de l'application linéaire Φ_U^V de la base de Horner à la base standard , qu'on peut également l'écrire par :

$$\mathrm{Bez}(U,V) = M_{(\Phi_U^V,\mathcal{B}_{Ho},\mathcal{B}_{St})}$$

Remarque 2.4. *Si $V(X)=1$ alors d'après la remarque 2.2,*

$$\mathrm{Bez}(U,1) = H_S = \mathrm{Hks}(u_1,\cdots,u_{n-1},u_n),$$

d'où :

$$\mathrm{Bez}(U,1)\mathrm{J}_n = \mathrm{Tops}(1,u_{n-1},\cdots u_1). \tag{2.3}$$

Comme les matrices $\mathrm{Bez}(U,V)$ et $\mathrm{H}(U,V)$ représentent la même application linéaire, on a alors le diagramme de décomposition suivant :

$$\begin{array}{ccc} & \mathrm{H}(U,V) & \\ B_{\mathrm{St}} & \longrightarrow & B_{\mathrm{Ho}} \\ \mathrm{Bez}(U,1)\ \uparrow & \curvearrowright & \downarrow\ \mathrm{Bez}(U,1) \\ B_{\mathrm{Ho}} & \longrightarrow & B_{\mathrm{St}} \\ & \mathrm{Bez}(U,V) & \end{array}$$

d'ou la formule qui relie les matrices de Bezout et de Hankel :

$$\mathrm{Bez}(U,V) = \mathrm{Bez}(U,1)\,\mathrm{H}(U,V)\,\mathrm{Bez}(U,1). \tag{2.4}$$

Puisque $\mathrm{Bez}(U,V)$ et $\mathrm{H}(U,V)$ sont deux matrices **congruente** et symétriques, on a (dans [2, 29]) les propriétés suivantes :

Propriétés

1. $\mathrm{rang}(\mathrm{Bez}(U,V)) = \mathrm{rang}(\mathrm{H}(U,V))$
2. $\mathrm{Sig}\,\mathrm{Bez}(U,V) = \mathrm{Sig}\,\mathrm{H}(U,V)$
3. $\deg(\mathrm{pgcd}(U,V)) = i \Longleftrightarrow \mathrm{rang}(\mathrm{H}(U,V)) = n-i.$
4. $\mathrm{Bez}(U,1)^{-1} = \mathrm{H}(U,1).$
5. $\mathrm{H}(U,V)\cdot\mathrm{Bez}(U,K) = \mathrm{I}$ où $V(X)\cdot K(X) = 1 \mod U(X)$.

Dans tous ce qui suit, nous désignerons par $U(X)$ et $V(X)$ deux polynômes **premiers entre eux** de $\mathbb{K}[X]$. Dans la section suivante on va voir comment on peut obtenir la liste des quotients et des restes de $U(X)$ et $V(X)$ à partir de la diagonalisation par bloc de $\mathrm{H}(U,V)$ et inversement.

2.2 Diagonalisation par bloc de $\mathrm{H}(U,V)$ et algorithme d'Euclide signé

On a vu dans le chapitre 1 que si nous itérons l'étape élémentaire prouvée dans le Théoréme 1.2, jusqu'au moment où l'on obtient comme matrice de Hankel restant à traiter une matrice Hankel-inférieure , nous obtenons ainsi une matrice diagonale par blocs « Hankel-inférieurs » D et une matrice triangulaire supérieure A telles que :

$$^{\mathrm{t}}AhA = D \in \mathcal{D}_n. \tag{2.5}$$

Cependant, on a prouvé que la détermination de A n'est pas exigé pour obtenir la matrice D, il suit du fait que les coefficients des quotients de la division X^{2n-1} par $\alpha_1 X^{2n-2}+\ldots+\alpha_{2n-1}$ dfinissent les blocs diagonaux.

2.2.1 Réduite diagonale par bloc de H(U,V) et suite des quotients

La matrice de Hankel h, associée à $U(X)$ et $V(X)$ est dans ce cas régulière et est définie par les $2n-1$ premiers coefficients de :

$$\mathcal{S}(X) = \frac{V(X)}{U(X)} = \sum_{i=1}^{\infty} h_i X^{-i},$$

autrement dit $h = \mathrm{H}(U,V) = \mathrm{Hk}(0,\ldots,0,h_{n-m},\ldots,h_{2n-1})$.
Si on applique le Théorème 1.2 à $\mathrm{H}(U,V)$, on obtient la décomposition suivante :

$$\boxed{^{\mathrm{t}}t_1^{-1}\mathrm{H}(U,V)t_1^{-1} = \begin{pmatrix} \mathrm{J}_{n-m}\mathrm{Bez}(q_1,1)\mathrm{J}_{n-m} & 0 \\ 0 & \mathrm{H}(V,r_1) \end{pmatrix},} \tag{2.6}$$

dans laquelle

$$U(X) = V(X)q_1(X) - r_1(X)(\deg(r_1(X)) < \deg(V(X))) \text{ et } t_1 = \mathrm{Tops}(h_{n-m},\ldots,h_{2n-m-1}).$$

En effet, pour avoir les coefficients des deux premiers blocs diagonaux de la réduite h' de h via le Théorème 1.2, on doit calculer les coefficients de la matrice inverse de $\mathrm{Tops}(h_{n-m},\ldots,h_{2n-1})$ qui seront donnés par les $n+m$ premiers coefficients de :

$$\begin{aligned} \frac{1}{\mathcal{S}(X)} = \frac{U(X)}{V(X)} &= q_1(X) - \frac{r_1(X)}{V(X)} \\ &= q_{1,n-m}X^{n-m} + q_{1,n-m-1}X^{n-m-1} + \ldots + q_{1,1}X + q_{1,0} + \sum_{i=1}^{\infty}\alpha_i X^{-i} \end{aligned}$$

De plus, d'après les équations 1.12 et 2.3 on peut ecrire :

$$h' = {}^{\mathrm{t}}t_1^{-1}\mathrm{H}(U,V)t_1^{-1} = \begin{pmatrix} h'_{11} & 0 \\ 0 & h'_{22} \end{pmatrix},$$

où

$$\begin{aligned} h'_{11} &= D_{11} = \mathrm{Hki}(q_{1,n-m},\ldots,q_{1,1}) = \mathrm{J}_{n-m}\mathrm{Bez}(q_1,1)\mathrm{J}_{n-m}, \\ h'_{22} &= -\mathrm{Hk}(\alpha_1,\ldots,\mu_m) = \mathrm{H}(V,r_1). \end{aligned}$$

D'où la preuve de l'équation 2.6.

A ce stade, remarquons le parallélisme entre cette première étape de réduction et celle de l'algorithme d'Euclide signé, en effet :

- Si on a la décomposition 2.6 de $\mathrm{H}(U,V)$, alors on a

$$\begin{aligned} n-m \;&=\; d_1 = \deg(q_1(X)),\\ q_1X &= q_{1,n-m}X^{n-m} + q_{1,n-m-1}X^{n-m-1} + \ldots + q_{1,1}X + q_{1,0} \end{aligned}$$

  et puisque on a $\mathrm{H}(V,r_1)$ on peut déduire $r_1(X)$ en appliquant la Proposition 2.1 et enfin on peut récupérer la constante $q_{1,0}$ de $q_1(X)$.
- Inversement, si on a $U(X) = V(X)q_1(X) - r_1(X)$ on peut alors calculer :
  - $\mathrm{Bez}(q_1,1)$ qui sera le premier bloc Hankel-inférieur de la réduite,
  - $\mathrm{H}(V,r_1)$ qui sera le deuxième bloc,
  - $t_1^{-1} = \mathrm{Tops}(q_{1,n-m},\ldots,q_{1,0},\alpha_1,\ldots,\alpha_{m-1})$,

d'où la formule 2.6 de décomposition de $\mathrm{H}(U,V)$.

Avec les notations de l'équation 2.2, l'Algorithme d'Euclide signé, appliqué à $U(X)$ et $V(X)$ génère la suite des quotients $\{q_i(X)\}$ et des restes $\{r_i(X)\}$ suivante :

$$\begin{cases} r_{-1}(X) := U(X), \quad r_0(X) := V(X)\\ r_{i-2}(X) := r_{i-1}(X)q_i(X) - r_i(X);\;\; \deg(r_i(X)) < \deg(r_{i-1}(X)),\; r_{k+1}(X) = 0;\; 1 \le i \le k+1 \end{cases} \tag{2.7}$$

Alors, si on itére l'étape précédente jusqu'au moment où l'on obtient comme matrice de Hankel restant à traiter une matrice Hankel-inférieure, on aura la réduite diagonale par blocs « Hankel-inférieurs » $D = \mathrm{Diag}(D_{11},\ldots,D_{k+1,k+1})$ de la Proposition 1.2, par conséquent on a :

$$\begin{aligned} &D_{ii} = \mathrm{J}_{d_i}\mathrm{Bez}(q_i,1)\mathrm{J}_{d_i};\; d_i = \deg(q_i(X)) = \deg(r_{i-2}(X)) - \deg(r_{i-1}(X)) : i = 1,\ldots,k;\\ &D_{k+1,k+1} = \mathrm{H}(r_{k-1},r_k) \;\text{ où }\; r_k(X) \in \mathbb{K} \;\text{ comme }\; \mathrm{pgcd}(U,V) = 1 : \qquad i = k+1; \end{aligned}$$

Ainsi, la suite des quotients générée par l'Algorithme d'Euclide signé appliqué à $U(X)$ et $V(X)$, fournit tous les blocs diagonaux « Hankel-inférieurs » de la réduite D et par conséquent il est inutile de calculer la matrice A pour obtenir D.

Remarque 2.5. *Remarquons que les matrices principales dominantes de* $\mathrm{H}(U,V)$ *sont* $H_{m_1}, H_{m_2}, \ldots, H_{m_{k+1}} = \mathrm{H}$*, d'ordre respectif* m_i*, avec :*

$$\begin{aligned} m_1 = d_1 = n - \deg(r_0(X)),\;\; m_2 = d_1 + d_2 = n - \deg(r_1(X)),\ldots,\\ m_i = d_1 + \cdots + d_i = n - \deg(r_{i-1}(X)),\ldots,\;\; m_{k+1} = n. \end{aligned} \tag{2.8}$$

Et grâce à l'équation 2.4, cette propriété est également vérifiée par la matrice $\mathrm{J}\,\mathrm{Bez}(U,V)\,\mathrm{J}$.

2.2.2 Exemple

Soit

$$U(X) = 6\,X^9 + 24\,X^8 + 44\,X^7 + 162\,X^6 + 60\,X^5 + 273\,X^4 + 32\,X^3 + 193\,X^2 - 70\,X - 10,$$

$$V(X) = 2\,x^7 + 6\,x^6 + 6\,x^5 + 40\,x^4 - 28\,x^3 + 65\,x^2 - 19\,x - 2.$$

On a alors,

$$\mathrm{H}(U,V) = \mathrm{Hk}\left(0, \frac{1}{3}, \frac{-1}{3}, \frac{-1}{9}, \frac{5}{9}, \frac{-11}{27}, \frac{-4}{9}, \frac{89}{81}, \frac{7}{81}, \frac{-584}{243}, \frac{-104}{243}, \frac{12667}{1458}, \frac{1018}{243}, \frac{-193345}{4374}, \frac{-36133}{2187}, \frac{3042241}{13122}, \frac{853193}{26244}\right).$$

Dans la suite, nous donnons le tableau (2.1) dans lequel nous faisons apparaître le parallélisme entre l'algorithme d'Euclide signé appliqué à $U(X)$ et $V(X)$ et notre méthode de diagonalisation par bloc appliquée à $\mathrm{H}(U,V)$.

Première étape : Diagonalisation en deux blocs de $\mathrm{H}(U,V)$:

$$
{}^{\mathrm{t}}t_1^{-1}\mathrm{H}(U,V)t_1^{-1} = \left(\begin{array}{cc|ccccccc} 0 & 3 & & & & & & & \\ 3 & 3 & & & & & & & \\ \hline & & 0 & 0 & -1 & 0 & 5 & -1 & -25 \\ & & 0 & -1 & 0 & 5 & -1 & -25 & 21/2 \\ & & -1 & 0 & 5 & -1 & -25 & 21/2 & 124 \\ & & 0 & 5 & -1 & -25 & 21/2 & 124 & -81 \\ & & 5 & -1 & -25 & 21/2 & 124 & -81 & -609 \\ & & -1 & -25 & 21/2 & 124 & -81 & -609 & 545 \\ & & -25 & 21/2 & 124 & -81 & -609 & 545 & \frac{11873}{4} \end{array}\right) = \begin{pmatrix} J_2\mathrm{Bez}(q_1,1)J_2 & O \\ O & \mathrm{H}(V,r_1) \end{pmatrix}
$$

L'algorithme d'Euclide signé appliqué à $U(X)$ et $V(X)$

$$
\begin{aligned} r_{-1}(X) &= r_0(X)q_1(X) - r_1(X) \\ &= r_0(X)(3X^2+3X+4) - (-2X^4-6X^3+4X^2-12X+2) \end{aligned}
$$

Deuxième étape : Diagonalisation en deux blocs de $\mathrm{H}(V,r_1)$

$$
{}^{\mathrm{t}}t_2^{-1}\mathrm{H}(V,r_1)t_2^{-1} = \left(\begin{array}{ccc|cccc} 0 & 0 & -1 & & & & \\ 0 & -1 & 0 & & & & \\ -1 & 0 & -5 & & & & \\ \hline & & & 0 & 1/2 & 0 & -1 \\ & & & 1/2 & 0 & -1 & 0 \\ & & & 0 & -1 & 0 & -3/2 \\ & & & -1 & 0 & -3/2 & 21/2 \end{array}\right) = \begin{pmatrix} J_3\mathrm{Bez}(q_2,1)J_3 & O \\ O & \mathrm{H}(r_1,r_2) \end{pmatrix}
$$

L'algorithme d'Euclide signé appliqué à $r_0(X)$ et $r_1(X)$

$$
\begin{aligned} r_0(X) &= r_1(X)q_2(X) - r_2(X) \\ &= r_1(X)(-X^3-5X+1) - (-X^2-3X+4) \end{aligned}
$$

Troisième étape : Diagonalisation en deux blocs de $\mathrm{H}(r_1,r_2)$

$$
{}^{\mathrm{t}}t_3^{-t}\mathrm{H}(r_1,r_2)t_3^{-1} = \left(\begin{array}{cc|cc} 0 & 2 & & \\ 2 & 0 & & \\ \hline & & 0 & -14 \\ & & -14 & 42 \end{array}\right) = \begin{pmatrix} J_2\mathrm{Bez}(q_3,1)J_2 & O \\ O & \mathrm{H}(r_2,r_3) \end{pmatrix}
$$

L'algorithme d'Euclide signé appliqué à $r_1(X)$ et $r_2(X)$

$$
\begin{aligned} r_1(X) &= r_2(X)q_3(X) - r_3(X) \\ &= r_2(X)(2X^2+4) - 14 \end{aligned}
$$

Tableau 2.1 – Parallélisme entre algorithme d'Euclide signé et diagonalisation par bloc de H(U,V).

Où

$$t_1 = \mathrm{Tops}\left(\frac{1}{3}, \frac{-1}{3}, \frac{-1}{9}, \frac{5}{9}, \frac{-11}{27}, \frac{-4}{9}, \frac{89}{81}, \frac{7}{81}, \frac{-584}{243}\right); \quad t_1^{-1} = \mathrm{Tops}(3,3,4,0,0,1,0,-5,1,);$$
$$t_2 = \mathrm{Tops}(-1,0,5,-1,-25,21/2,124); \qquad t_2^{-1} = \mathrm{Tops}(-1,0,-5,1,0,-1/2,0,);$$
$$t_3 = \mathrm{Tops}(1/2,0,-1,0); \qquad t_3^{-1} = \mathrm{Tops}(2,0,4,0).$$

Si on pose

$$A = t_1^{-1}\mathrm{Diag}(I_2, t_2^{-1})\mathrm{Diag}(I_5, t_3^{-1}) = \begin{pmatrix} 3 & 3 & -4 & 0 & -20 & 6 & 0 & 8 & 0 \\ 0 & 3 & -3 & -4 & -15 & -34 & 6 & -71 & 8 \\ 0 & 0 & -3 & -3 & -19 & -24 & -34 & -45 & -71 \\ 0 & 0 & 0 & -3 & -3 & -38 & -24 & -110 & -45 \\ 0 & 0 & 0 & 0 & -3 & -6 & -38 & -36 & -110 \\ 0 & 0 & 0 & 0 & 0 & -6 & -6 & -50 & -36 \\ 0 & 0 & 0 & 0 & 0 & 0 & -6 & -6 & -50 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & -6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 \end{pmatrix},$$

alors, la réduite diagonale par bloc de $h = \mathrm{H}(U,V)$ est :

$$D = {}^{\mathrm{t}}A\mathrm{H}(U,V)A = \begin{pmatrix} \begin{pmatrix} 0 & 3 \\ 3 & 3 \end{pmatrix} & & & & \\ & \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & -5 \end{pmatrix} & & & \\ & & \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix} & & \\ & & & \begin{pmatrix} 0 & -14 \\ -14 & 42 \end{pmatrix} & \end{pmatrix}.$$

2.2.3 Matrice de passage et suite des restes

On a vu dans la section 2.2.1 qu'étant donnée la suite des quotients, il est inutile de calculer la matrice A pour obtenir D telle que $D = {}^{\mathrm{t}}A\mathrm{H}(U,V)A$. La question naturelle qui se pose maintenant est : est-il aussi possible d'obtenir A à partir de la division de polynômes ? et la réponse est oui, en effet :

d'après l'exemple 2.2.2 précédent, on constate que la suite des restes de $U(X)$ et $V(X)$ apparaît dans la matrice :

$$A^{-1}\mathrm{Bez}(U,1)J = \begin{pmatrix} 2 & 6 & 6 & 40 & -28 & 65 & -19 & -2 & 0 \\ 0 & 2 & 6 & 6 & 40 & -28 & 65 & -19 & -2 \\ 0 & 0 & -2 & -6 & 4 & -12 & 2 & 0 & 0 \\ 0 & 0 & 0 & -2 & -6 & 4 & -12 & 2 & 0 \\ 0 & 0 & 0 & 0 & -2 & -6 & 4 & -12 & 2 \\ 0 & 0 & 0 & 0 & 0 & -1 & -3 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix},$$

en effet, puisque

$$A = t_1^{-1}\mathrm{Diag}(I_2, t_2^{-1})\mathrm{Diag}(I_5, t_3^{-1})$$

alors

$$\begin{aligned} A^{-1}\mathrm{Bez}(U,1)J &= \mathrm{Diag}(\mathrm{I}_5, t_3)\mathrm{Diag}(\mathrm{I}_2, t_2)t_1 \underbrace{\mathrm{Bez}(U,1)J} \\ &\overset{(2.4)}{=} \mathrm{Diag}(\mathrm{I}_5, t_3)\mathrm{Diag}(\mathrm{I}_2, t_2)t_1\mathrm{Tops}(u_9, \cdots, u_1) \end{aligned}.$$

Comme la multiplication des matrices Toeplitz-supérieures représente une multiplication de polynômes et comme la matrice
t_1 est
- une matrice Toeplitz-supérieure d'ordre 9 où $9 = \deg(U(X))$.
- définie par les 9 premiers éléments à compter à partir du premier élément non nuls dans la liste qui code $\mathrm{H}(U,V) \simeq V(X)/U(X)$

on a alors l'identité suivante :

$$t_1\,\mathrm{Tops}(u_9, \ldots, u_1) = \mathrm{Tops}(\mathbf{XV}).$$

De même, comme la matrice
t_2 est
- une matrice Toeplitz-supérieure d'ordre 7 où $7 = \deg(U(X)) - d_1$, $d_1 = \deg(q_1(X))$.
- définie par les 7 premiers éléments à compter à partir du premier élément non nuls dans la liste qui code $\mathrm{H}(V, r_1) \simeq r_1(X)/V(X)$

on a alors l'identité suivante :

$$\mathrm{Diag}(\mathrm{I}_2, t_2)\mathrm{Tops}(\mathbf{XV}) = \begin{pmatrix} \text{coefficients de V(X)} \\ \text{coefficients de V(X)} \\ \text{coefficients de } r_1(X) \\ \text{coefficients de } r_1(X) \\ \text{coefficients de } r_1(X) \\ O_{5\times 5} \quad \mathrm{Tops}(\mathbf{X^2 r}_1) \end{pmatrix} \begin{matrix} \Big\} d_1 = 2 \\ \Big\} d_2 = 3 \end{matrix}$$

De même, comme la matrice
t_3 est
- une matrice Toeplitz-supérieure d'ordre 4 où $4 = \deg(U(X)) - (d_1 + d_2)$, $d_2 = \deg(q_2(X))$.

– définie par les 4 premiers éléments à compter à partir du premier élément non nul dans la liste qui code $\mathrm{H}(r_1, r_2) \simeq r_2(X)/r_1(X)$

on a alors l'identité suivante :

$$\mathrm{Diag}(\mathrm{I}_5, t_3)\mathrm{Diag}(\mathrm{I}_2, t_2)\mathrm{Tops}(\mathbf{XV}) = \begin{pmatrix} \text{coefficients de V(X)} & \\ \quad \text{coefficients de V(X)} & \\ \quad\quad \text{coefficients de } r_1(X) & \\ \quad\quad\quad \text{coefficients de } r_1(X) & \\ \quad\quad\quad\quad \text{coefficients de } r_1(X) & \\ \quad\quad\quad\quad\quad \text{coefficients de } r_2(X) & \\ \quad\quad\quad\quad\quad\quad \text{coefficients de } r_2(X) & \\ O_{7\times 7} \qquad \mathrm{Tops}(\mathbf{Xr}_2) & \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\} d_1 = 2 \\ \left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\} d_2 = 3 \\ \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\} d_3 = 2 \\ \end{matrix}$$

d'où le résultat.

Généralisation

Reprenons les notations de la section 2.2.1 et supposons que la réduite D de $\mathrm{H}(U, V)$ aura $k+1$ blocs diagonal Hankel-inférieurs, $D = {}^{\mathrm{t}}A\mathrm{H}(U, V)A = diag(D_{11}, D_{22}, \ldots, D_{kk}, D_{k+1,k+1})$, où

$$A = t_1^{-1}\mathrm{Diag}(\mathrm{I}_{m_1}, t_2^{-1}) \ldots \mathrm{Diag}(\mathrm{I}_{m_{k-1}}, t_k^{-1}),$$

dans laquelle :

t_1 est

– une matrice Toeplitz-supérieure d'ordre $n = \deg(U(X))$,

– définie par les n premiers éléments à compter à partir du premier élément non nuls de la liste qui code $\mathrm{H}(U, V)$, c'est-à-dire les n premier coefficients de la série $V(X)/U(X)$.

t_2 est

– une matrice Toeplitz-supérieure d'ordre $n-d_1 = n-m_1 = \deg(V(X))$ où $d_1 = \deg(q_1(X))$,

– définie par les m premiers éléments à compter à partir du premier élément non nuls de la liste qui code $\mathrm{H}(V, r_1)$, c'est-à-dire les m premier coefficients de la série $r_1(X)/V(X)$.

et ainsi de suite jusqu'au dernière matrice t_k.

plus généralement, si on pose $m_0 = 0$ on a pour tout $i = 1, ..., k$ la matrice :

t_i est

– une matrice Toeplitz-supérieure d'ordre $n - (d_1 + \ldots + d_{i-1}) = n - m_{i-1} = \deg(r_{i-2}(X))$ où $d_{i-1} = \deg(q_{i-1}(X))$,

– définie par les $n - m_{i-1}$ premiers éléments à compter à partir du premier élément non nul dans la liste qui code $\mathrm{H}(r_{i-2}, r_{i-1}) \simeq r_{i-1}(X)/r_{i-2}(X)$.

Le lemme suivant montre que $A^{-1}\mathrm{Bez}(U, 1)\mathrm{J}$ est une matrice triangulaire supérieure dont les lignes sont données par la liste des restes $\{r_0(X), \ldots, r_{k-1}(X)\}$ donnée par l'algorithme d'Euclide signé appliqué à $U(X)$ et $V(X)$.

Lemme 2.2.1. *Supposons que chaque reste $r_i(X)$ est écrit comme suis :*

$$r_i(X) = r_{i,n-m_{i+1}}X^{n-m_{i+1}} + \cdots + r_{i,0}$$

.

Alors la matrice $A^{-1}\mathrm{Bez}(U,1)\mathrm{J}$ *est donnée par :*

$$\begin{pmatrix}
v_m & \cdots & \cdots & \cdots & \cdots & v_0 & & \\
\ddots & & & & & \ddots & & \\
 & v_m & \cdots & \cdots & \cdots & \cdots & v_0 & \\
 & r_{1,n-m_2} & \cdots & \cdots & r_{1,0} & & & \\
 & & \ddots & & & \ddots & & \\
 & & & r_{1,n-m_2} & \cdots & \cdots & r_{1,0} & \\
 & & & \ddots & & & & \\
 & & & & r_{k-3,n-m_{k-2}} & \cdots & r_{k-3,0} & \\
 & & & & & \ddots & \ddots & \\
 & & & & r_{k-3,n-m_{k-2}} & \cdots & r_{-3,0} & \\
 & & & & & & & \mathrm{Tops}(\mathbf{X^{d_{k-1}-1}r}_{k-2})
\end{pmatrix}$$

Démonstration. Puisque

$$A \;=\; t_1^{-1}\mathrm{Diag}(\mathrm{I}_{m_1}, t_2^{-1})\ldots\mathrm{Diag}(\mathrm{I}_{m_{k-1}}, t_k^{-1}),$$

alors

$$\begin{aligned}
A^{-1}\mathrm{Bez}(U,1)J &= \mathrm{Diag}(\mathrm{I}_{m_{k-1}}, t_k)\ldots\mathrm{Diag}(\mathrm{I}_{m_1}, t_2)t_1\underbrace{\mathrm{Bez}(U,1)J}\\
&\overset{(2.4)}{=} \mathrm{Diag}(\mathrm{I}_{m_{k-1}}, t_k)\ldots\mathrm{Diag}(\mathrm{I}_{m_1}, t_2)t_1\mathrm{Tops}(u_n,\cdots,u_1)
\end{aligned}.$$

Comme la multiplication des matrices Toeplitz-supérieures représente une multiplication de polynômes et comme la matrice t_1 est une matrice Toeplitz-supérieure d'ordre n, définie par les n premiers éléments de la série formelle $V(X)/U(X)$; on a alors l'identité suivante :

$$t_1\,\mathrm{Tops}(u_n,\ldots,u_1) = \mathrm{Tops}(\mathbf{X^{n-m-1}V}).$$

De même, la matrice t_2 est une matrice Toeplitz-supérieure d'ordre $n-m_1$, définie par les $n-m_1$ premiers éléments de la série formelle $r_1(X)/V(X)$; on a alors

$$\mathrm{Diag}(I_{m_1}, t_2)\,\mathrm{Tops}(\mathbf{X^{n-m-1}V}) = \begin{pmatrix}
v_m & \cdots & v_0 & & \\
 & \ddots & & \ddots & \\
 & v_m & \cdots & v_0 & \\
 & & & & \\
 & & & & \mathrm{Tops}(\mathbf{X^{d_2-1}r}_1)
\end{pmatrix}.$$

Finalement, puisque chaque matrice t_i est une matrice Toeplitz-supérieure d'ordre $n-m_{i-1}$, définie par les $n-m_{i-1}$ premiers termes de la série formelle $r_{i-1}(X)/r_{i-2}(X)$, nous pouvons suivre le même raisonnement avec les autres produits, d'où

$$\begin{aligned}
A^{-1}\mathrm{Bez}(U,1)J &\overset{(2.4)}{=} \mathrm{Diag}(\mathrm{I}_{m_{k-1}}, t_k)\ldots\mathrm{Diag}(\mathrm{I}_{m_2}, t_3)\mathrm{Diag}(\mathrm{I}_{m_1}, t_2)\,\underbrace{t_1\mathrm{Tops}(u_n,\cdots,u_1)}\\
&= \mathrm{Diag}(\mathrm{I}_{m_{k-1}}, t_k)\ldots\mathrm{Diag}(\mathrm{I}_{m_2}, t_3)\mathrm{Diag}(\mathrm{I}_{m_1}, t_2)\;\;\underbrace{\mathrm{Tops}(\mathbf{X^{d_1-1}V})}
\end{aligned}$$

$$= \mathrm{Diag}(\mathrm{I}_{m_{k-1}}, t_k) \dots \mathrm{Diag}(\mathrm{I}_{m_2}, t_3)\, \mathrm{Diag}(\mathrm{I}_{m_1}, t_2)) \underbrace{\begin{pmatrix} v_m \dots\dots & & v_0 & \\ & \ddots & & \ddots \\ & & v_m & \dots\dots v_0 \\ & & & v_m \quad \dots v_1 \\ & & & \ddots \quad \vdots \\ & & & \ddots \vdots \\ & & & v_m \end{pmatrix}}_{\begin{matrix} \} d_1 \\ \\ \\ \\ \\ \\ \} \mathrm{Tops}(v_m, \dots, v_1) \end{matrix}}$$

$$= \mathrm{Diag}(\mathrm{I}_{m_{k-1}}, t_k) \dots \mathrm{Diag}(\mathrm{I}_{m_2}, t_3) \underbrace{\begin{pmatrix} v_m \dots\dots\dots v_0 & & \\ \ddots & & \ddots \\ & v_m \dots\dots\dots v_0 & \\ & & \\ & & \mathrm{Tops}(\mathbf{X^{d_2-1}r_1}) \end{pmatrix}}_{\} d_1}$$

$$= \mathrm{Diag}(\mathrm{I}_{m_{k-1}}, t_k) \dots \underbrace{\begin{pmatrix} v_m \dots\dots\dots & \dots\dots & v_0 & \\ \ddots & & & \ddots \\ v_m \dots & \dots\dots & \dots \dots & v_0 \\ r_{1,n-m_2} \dots\dots & & r_{1,0} & \\ & \ddots & & \ddots \\ & & r_{1,n-m_2} \dots \dots & r_{1,0} \\ & & & \\ & & & \mathrm{Tops}(\mathbf{X^{d_3-1}r_2}) \end{pmatrix}}_{\begin{matrix} \} d_1 \\ \\ \} d_2 \\ \\ \\ \\ \\ \\ \\ \end{matrix}}$$

$$= \begin{pmatrix} v_m \dots\dots & \dots & \dots\dots & \dots & & \dots & & v_0 & & \\ \ddots & & & & & & & & \ddots & \\ & v_m & \dots & \dots\dots & \dots & & \dots & & \dots & v_0 \\ & r_{1,n-m_2} & \dots\dots & \dots & & r_{1,0} & & & & \\ & & \ddots & & & & & & \ddots & \\ & & & r_{1,n-m_2} & & \dots & & & \dots & r_{1,0} \\ & & & \ddots & & & & & & \\ & & & & r_{k-3,n-m_{k-2}} & & \dots & & r_{k-3,0} & \\ & & & & & \ddots & & & & \ddots \\ & & & & & r_{k-3,n-m_{k-2}} & & & \dots & r_{-3,0} \\ & & & & & & & & & \\ & & & & & & & \mathrm{Tops}(\mathbf{X^{d_{k-1}-1}r_{k-2}}) & & \end{pmatrix} \begin{matrix} \} d_1 \\ \\ \\ \} d_2 \\ \\ \\ \vdots \\ \} d_{k-2} \\ \\ \\ \\ \end{matrix}$$

□

Remarque 2.6. *Dans cette section on a considéré deux polynômes $U(X)$ et $V(X)$ premiers entre eux. Ci c'est pas le cas, la seule différence est que* H(U,V) *n'est pas régulière. Donc, on obtient par le même raisonnement qu'auparavant, la réduite diagonale par blocs « Hankel-inférieurs » $D = \mathrm{Diag}(D_{11},\ldots,D_{k+1,k+1})$ tels que :*

$$D_{ii} = \mathrm{J\,Bez}(q_i,1)\,\mathrm{J}, \quad i=1,\ldots,k;$$
$$D_{k+1,k+1} = \textit{ bloc nul }, i = k+1.$$

2.3 Diagonalisation par bloc de J Bez(U,V) J

Dans cette section nous présentons la méthode de diagonalisation par blocs de la matrice J Bez(U,V) J, décrite dans [7]. Dans cette diagonalisation les auteurs utilisent la notion de *complément de Schur* d'une matrice.

Définition 2.3.1. *Soient A,B,C et D quatre matrices d'ordre respectif $p\times p$, $p\times q$, $q\times p$ et $q\times q$, avec A régulière. Posons*

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

alors M est une matrice carrée d'ordre $(p+q)\times(p+q)$.
On appelle complément de Schur *du bloc A dans M, la matrice carrée d'ordre $p\times p$:*

$$S = D - C\,A^{-1}\,B.$$

2.3.1 Réduite diagonale par bloc de J Bez(U,V) J

Étant donné m_i (voir l'équation (2.8)), le mineur $(\mathrm{J\,Bez}(U,V)\,\mathrm{J})_{m_i}$ est non nul, comme nous l'avons dit dans la remarque 2.5. Ci-après nous présentons l'algorithme parallèle décrit dans [7]), pour la diagonalisation par blocs de JBez(U,V)J.

Proposition 2.2.

$$\mathrm{JBez}(U,V)\mathrm{J} = \begin{pmatrix} (\mathrm{JBez}(U,V)\mathrm{J})_{m_i} & {}^{\mathrm{t}}M \\ M & N \end{pmatrix}$$
$$= \begin{pmatrix} T & \mathrm{O} \\ M(\mathrm{JBez}(U,V)\mathrm{J})_{m_i}^{-1}T & I \end{pmatrix} \begin{pmatrix} \mathrm{JBez}(U^{(i)},V^{(i)})\mathrm{J} & \mathrm{O} \\ \mathrm{O} & \mathrm{JBez}(r_{i-1},r_i)\mathrm{J} \end{pmatrix} \begin{pmatrix} {}^{\mathrm{t}}T & {}^{\mathrm{t}}T(\mathrm{JBez}(U,V)\mathrm{J})_{m_i}^{-1}\,{}^{\mathrm{t}}M \\ \mathrm{O} & I \end{pmatrix}, \tag{2.9}$$

où

$$T = \mathrm{Topi}(u_n,\ldots,u_{n-m_1+1})\mathrm{Bez}(U^{(i)},1)^{-1}\mathrm{J},$$

les polynômes $U^{(i)}(X), V^{(i)}(X)$ vérifient

$$H(U,V)_{m_i} = \mathrm{H}(U^{(i)},V^{(i)})$$

et

$$\mathrm{JBez}(r_{i-1},r_i)\mathrm{J} = N - M(\mathrm{JBez}(U,V)\mathrm{J})_{m_i}^{-1}\,{}^{\mathrm{t}}M$$

qui est égal au complement de Schur de $(\mathrm{JBez}(U,V)\mathrm{J})_{m_i}$ dans $(\mathrm{JBez}(U,V)\mathrm{J})$.

Nous avons séquentiellement appliqué l'identité (2.9) dans le but de la comparer avec la nôtre. Nous commençons à calculer le premier bloc diagonal.

Désignons par D_b la matrice diagonale par bloc LU-quivalente à la matrice $\mathrm{JBez}(U,V)\mathrm{J}$, et par $D_{b_{11}}$ son premier bloc. Alors

$$m_1 = n - m,\ h_{11} = \mathrm{H}(U,V)_{n-m} = \mathrm{Hki}(h_{n-m}, \ldots, h_{2n-2m-1})$$

et via la remarque 2.3, on peux choisir $V^{(1)}(X) = 1$ de telle sorte que

$$h_{11} = H(U,V)_{n-m} = H(U^{(1)}, 1).$$

Comme

$$\mathrm{H}(U^{(1)}, 1)^{-1} = \mathrm{Bez}(U^{(1)}, 1),$$

on a

$$\mathrm{J}h_{11}^{-1}\mathrm{J} = \mathrm{JH}(U^{(1)}, 1)^{-1}\mathrm{J} = \mathrm{JBez}(U^{(1)}, 1)\mathrm{J}.$$

De plus, l'équation (2.6) nous donne

$$\mathrm{J}h_{11}^{-1}\mathrm{J} = \mathrm{JBez}(q_1, 1)\mathrm{J}.$$

Alors

$$D_{b_{11}} = \mathrm{JBez}(U^{(1)}, 1)\mathrm{J} = \mathrm{JBez}(q_1, 1)\mathrm{J}.$$

De cette façon nous obtenons récursivement une matrice supérieure A_b et une matrice diagonale par blocs D_b telle que :

$$^{\mathrm{t}}A_b\, \mathrm{JBez}(U,V)\mathrm{J}\, A_b = D_b \tag{2.10}$$

Pour conclure, nous pouvons affirmer que les blocs de D et D_b sont les mêmes à l'exception du dernier,

$$D_{kk} = \mathrm{H}(r_{k-2}, r_{k-1}), \quad D_{b_{kk}} = \mathrm{J}\,\mathrm{Bez}(r_{k-2}, r_{k-1})\,\mathrm{J}.$$

2.3.2 La matrice A_b

Étant donné la suite des quotients signés $\{q_i(X)\}$ de $U(X)$ et $V(X)$, il est donc inutile de calculer la matrice A_b afin d'obtenir D_b.

Comme nous l'avons expliquer dans la section 2.2.3, il est aussi possible d'obtenir directement A_b^{-1} à partir de la suite des restes signés $\{r_i(X)\}$ de $U(X)$ et $V(X)$. En effet, il suit du lemme 2.2.1 et de l'équation (2.1) que :

$$A_b^{-1} = \begin{pmatrix}
v_m & \cdots & \cdots & \cdots & & v_0 & & \\
\ddots & & & & & & \ddots & \\
 & v_m & \cdots & \cdots & \cdots & \cdots & & v_0 \\
 & & r_{1,n-m_2} & \cdots & \cdots & & r_{1,0} & \\
 & & & \ddots & & & & \ddots \\
 & & & & r_{1,n-m_2} & \cdots & \cdots & r_{1,0} \\
 & & & & \ddots & & & \\
 & & & & & r_{k-2,n-m_{k-1}} & \cdots & r_{k-2,0} \\
 & & & & & \ddots & \ddots & \\
 & & & & & r_{k-2,n-m_{k-1}} & \cdots & r_{k-2,0} \\
 & & & & & & & \\
 & & & & & & & \mathrm{I}_{\mathrm{d}_k}
\end{pmatrix}$$

Remarque 2.7. *Si $H(U,V)$ est singulière alors $D = D_b$.*

Remarque 2.8. *Les résultats susdites décrivent des algorithmes de diagonalisation par blocs des matrices de Hankel et Bezout, associées à deux polynômes. Cependant une telle diagonalisation n'est pas unique. Par exemple, observons comment est-il possible d'obtenir une diagonalisation par blocs de* H(U,V) *à partir de celle de* JBez(U,V)J *et vice versa.*

Si

$$ {}^{\mathrm{t}}A_b \mathrm{JBez}(U,V)\mathrm{J}A_b = D_b $$

alors

$$ \left({}^{\mathrm{t}}A_b \mathrm{JBez}(U,1)\right) \mathrm{H}(U,V) \left(\mathrm{Bez}(U,1)\mathrm{J}A_b\right) = D_b. $$

Et si

$$ {}^{\mathrm{t}}A\mathrm{H}(U,V)A = D $$

alors

$$ \left({}^{\mathrm{t}}A\mathrm{Bez}(U,1)^{-1}\mathrm{J}\right) \mathrm{JBez}(U,V)\mathrm{J} \left(\mathrm{JBez}(U,1)^{-1}A\right) = D. $$

De toute façon, si nous avons plusieurs matrices supérieures vérifiant l'équation 2.5 (respectivement l'équation 2.10) et si nous faisons la diagonale principale égale à 1 *alors nous devons trouver la même $m_i^{\text{ème}}$ colonne dans toutes ces matrices (voir le Théorème 1.1).*

Remarque 2.9. *Notons que dans l'algorithme d'Euclide, $q_{k+1}(X) = \frac{r_{k-1}(X)}{r_k(X)}$, par conséquent la suite des quotients signés $\{q_i(X) = q_{i,d_i}X^{d_i} + q_{i,d_i-1}X^{d_i-1} + \ldots + q_{i,1}X + q_{i,0}; \quad i = 1,\ldots,k\}$ fournit l'expression de la signature,*

$$ \mathrm{Sig}(\mathrm{Bez}(U,V)) = \mathrm{Sig}(\mathrm{H}(U,V)) = \sum_{i=n}^{k} \begin{cases} 0, & \text{si } d_i = \deg(q_i(X)) \text{ est pair;} \\ \mathrm{sig}(q_{i,d_i}), & \text{si } d_i = \deg(q_i(X)) \text{ est impair.} \end{cases} \tag{2.11} $$

Ainsi, nos conclusions peuvent être utilisées pour simplifier beaucoup de preuves de résultats concernant la signature d'une matrice de Hankel :

- *par exemple, dans [20], l'auteur prouve l'identité (2.11) d'une manière détaillée et non pas d'une façon directe de la diagonalisation par blocs.*
- *Dans [27], le Théorème 3.4 se trouve être un corollaire de Théorème 1.2.*
- *Dans [4], nous présentons une preuve simple pour le Théorème de Frobenius qui caractérise la signature d'une matrice Hankel par le signe de ses mineurs principaux dominants.*

D'autre part, il y a d'autres applications intéressantes liées au domaine de la Géométrie Algébrique Réelle, comme le calcul des racines réelles d'un polynôme réel ou le problème Routh-Hurwitz pour des polynômes réels (voir [3] et [26]).

2.3.3 Exemple

Nous reprenons ici l'exemple 2.2.2. On a alors,

$$\mathrm{J}\,\mathrm{Bez}(u,v)\,\mathrm{J} = \begin{pmatrix} 0 & 12 & 36 & 36 & 240 & -168 & 390 & -114 & -12 \\ 12 & 84 & 180 & 384 & 792 & -282 & 1446 & -468 & -48 \\ 36 & 180 & 324 & 936 & 932 & 150 & 2006 & -744 & -68 \\ 36 & 384 & 936 & 1544 & 4992 & -2722 & 8628 & -2726 & -264 \\ 240 & 792 & 932 & 4992 & -1960 & 6756 & 16 & -984 & -60 \\ -168 & -282 & 150 & -2722 & 6756 & -8908 & 9041 & -2447 & -146 \\ 390 & 1446 & 2006 & 8628 & 16 & 9041 & 5037 & -2714 & -344 \\ -114 & -468 & -744 & -2726 & -984 & -2447 & -2714 & 539 & 264 \\ -12 & -48 & -68 & -264 & -60 & -146 & -344 & 264 & -50 \end{pmatrix}.$$

Dans le tableau (2.2) nous faisons apparaître le parallélisme entre l'algorithme d'Euclide signé appliqué à $U(X)$ et $V(X)$ et la diagonalisation par bloc de $\mathrm{JBez}(U,V)\mathrm{J}$, dans lequel les matrices b_i sont calculées récursivement via l'identité 2.9 :

Première étape : Diagonalisation en deux blocs de JBez(U,V)J :

$$
{}^{\mathrm{t}}b_1^{-1}\mathrm{JBez}(U,V)\mathrm{J}b_1^{-1} = \left(\begin{array}{cc|ccccccc}
0 & 3 & & & & & & & \\
3 & 3 & & & & & & & \\
\hline
 & & 0 & 0 & -4 & -12 & 8 & -24 & 4 \\
 & & 0 & -4 & -24 & -28 & 0 & -68 & 12 \\
 & & -4 & -24 & -40 & -36 & -44 & -60 & 12 \\
 & & -12 & -28 & -36 & -340 & 230 & -506 & 76 \\
 & & 8 & 0 & -44 & 230 & -228 & 298 & -68 \\
 & & -24 & -68 & -60 & -506 & 298 & -772 & 138 \\
 & & 4 & 12 & 12 & 76 & -68 & 138 & -62
\end{array}\right) = \begin{pmatrix} \mathrm{JBez}(q_1,1)\mathrm{J} & O \\ O & \mathrm{JBez}(V,r_1)\mathrm{J} \end{pmatrix}
$$

L'algorithme d'Euclide signé appliqué à $U(X)$ et $V(X)$
$r_{-1}(X) = r_0(X)q_1(X) - r_1(X)$
$\quad = r_0(X)(3X^2+3X+4) - (-2X^4-6X^3+4X^2-12X+2)$

Deuxième étape : Diagonalisation en deux blocs de JBez(V,r_1)J

$$
{}^{\mathrm{t}}b_2^{-1}\mathrm{JBez}(V,r_1)\mathrm{J}b_2^{-1} = \left(\begin{array}{ccc|cccc}
0 & 0 & -1 & & & & \\
0 & -1 & 0 & & & & \\
-1 & 0 & -5 & & & & \\
\hline
 & & & 0 & 2 & 6 & -8 \\
 & & & 2 & 12 & 10 & -24 \\
 & & & 6 & 10 & -48 & 18 \\
 & & & -8 & -24 & 18 & -42
\end{array}\right) = \begin{pmatrix} \mathrm{JBez}(q_2,1)\mathrm{J} & O \\ O & \mathrm{JBez}(r_1,r_2)\mathrm{J} \end{pmatrix}
$$

L'algorithme d'Euclide signé appliqué à $r_0(X)$ et $r_1(X)$
$r_0(X) = r_1(X)q_2(X) - r_2(X)$
$\quad = r_1(X)(-X^3-5X+1) - (-X^2-3X+4)$

Troisième étape : Diagonalisation en deux blocs de JBez(r_1,r_2)J

$$
{}^{\mathrm{t}}b_3^{-1}\mathrm{JBez}(r_1,r_2)\mathrm{J}b_3^{-1} = \left(\begin{array}{cc|cc}
0 & 2 & & \\
2 & 0 & & \\
\hline
 & & 0 & -14 \\
 & & -14 & -42
\end{array}\right) = \begin{pmatrix} J_2\mathrm{Bez}(q_3,1)J_2 & O \\ O & \mathrm{JBez}(r_2,r_3)\mathrm{J} \end{pmatrix}
$$

L'algorithme d'Euclide signé appliqué à $r_1(X)$ et $r_2(X)$
$r_1(X) = r_2(X)q_3(X) - r_3(X)$
$\quad = r_2(X)(2X^2+4) - 14$

Tableau 2.2 – Parallélisme entre algorithme d'Euclide signé et diagonalisation par bloc de Bez(U,V).

Avec

$$b_1 = \begin{pmatrix} 2 & 6 & 6 & 40 & -28 & 65 & -19 & -2 & 0 \\ 0 & 2 & 6 & 6 & 40 & -28 & 65 & -19 & -2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

$$b_2 = \begin{pmatrix} -1/2 & 3/2 & -11/2 & -45 & 41 & -69 & 11 \\ 0 & -1/2 & 3/2 & 11 & -12 & 19 & -3 \\ 0 & 0 & -1/2 & -3 & 2 & -6 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}; b_3 = \begin{pmatrix} -1 & 3 & 13 & -12 \\ 0 & -1 & -3 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Par conséquent, si nous posons

$$A_b = b_1^{-1} \operatorname{Diag}(I_2, b_2^{-1}) \operatorname{Diag}(I_5, b_3^{-1}) = \begin{pmatrix} \frac{1}{2} & \frac{-3}{2} & -3 & \frac{29}{2} & \frac{-173}{2} & \frac{1375}{2} & \frac{-5391}{2} & -11931 & 10952 \\ 0 & \frac{1}{2} & \frac{3}{2} & -3 & 22 & -176 & \frac{1375}{2} & 3046 & -2793 \\ 0 & 0 & \frac{-1}{2} & \frac{3}{2} & \frac{-11}{2} & 45 & -176 & -777 & 715 \\ 0 & 0 & 0 & \frac{-1}{2} & \frac{3}{2} & -11 & 45 & 198 & -183 \\ 0 & 0 & 0 & 0 & \frac{-1}{2} & 3 & -11 & -51 & 45 \\ 0 & 0 & 0 & 0 & 0 & -1 & 3 & 13 & -12 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

on aura alors

$$D_b = {}^{\mathrm{t}}A_b\,\mathrm{J}\,\mathrm{Bez}(U,V)\,\mathrm{J}\,A_b = \begin{pmatrix} 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -14 \\ 0 & 0 & 0 & 0 & 0 & 0 & -14 & -42 \end{pmatrix}.$$

De plus comme nous l'avons dit dans la section 2.3.2, remarquons que les restes apparaissent dans la matrice suivante :

$$A_b^{-1} = \begin{pmatrix} 2 & 6 & 6 & 40 & -28 & 65 & -19 & -2 & 0 \\ 0 & 2 & 6 & 6 & 40 & -28 & 65 & -19 & -2 \\ 0 & 0 & -2 & -6 & 4 & -12 & 2 & 0 & 0 \\ 0 & 0 & 0 & -2 & -6 & 4 & -12 & 2 & 0 \\ 0 & 0 & 0 & 0 & -2 & -6 & 4 & -12 & 2 \\ 0 & 0 & 0 & 0 & 0 & -1 & -3 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

3. Variantes de l'algorithme de Berlekamp–Massey

Introduction

Nous revisitons dans ce chapitre l'algorithme de Berlekamp-Massey , qui calcule le polynôme générateur d'une suite récurrente linéaire à partir des premiers termes de la suite.

L'algorithme classique est présenté dans la section 3.1. Une variante, légèrement plus simple est donnée dans la section 3.2. L'explication de l'algorithme est également plus facile. En outre cette nouvelle version autorise une approche « dynamique » utile dans certaines situations, expliquée dans la section 3.4.

La variante exposée dans la section 3.2 a directement été inspirée par l'algorithme de diagonalisation que nous avons présenté au chapitre 1. Profitant de l'accélération des calculs que nous avons établie pour les algorithmes les plus sophistiqués du chapitre 1 nous en déduisons un algorithme « à la Berlekamp-Massey » un peu plus rapide que l'algorithme usuel. Le coût en nombre d'opérations arithmétiques pour la suite des quotients est ainsi diminué d'un tiers environ. Pour le coût total il est diminué d'un quart.

Les résultats de ce chapitre sont en partie exposés en anglais dans l'article [5].

3.1 L'algorithme de Berlekamp-Massey usuel

3.1.1 Suites récurrentes linéaires

On considère une suite $\underline{a} = (a_k)_{k\in\mathbb{N}}$ d'éléments de $\mathbb{K}$ et un entier $n \in \mathbb{N}$.

Une *relation récurrente linéaire d'ordre n* pour cette suite est définie par la donnée de $n+1$ éléments $c_0, c_1, \ldots, c_n$ $(c_n \neq 0)$ de $\mathbb{K}$ vérifiant :

$$\forall k \geq 0 \quad c_0 a_k + c_1 a_{k+1} + \cdots + c_n a_{k+n} = 0 \tag{3.1}$$

Le polynôme $G(X) = \sum_{i=0}^{n} c_i X^i$ est alors appelé *un polynôme générateur* de la suite $\underline{a}$.

Définition 3.1.1. *On appelle* suite récurrente linéaire*, toute suite* $\underline{a}$ *pour laquelle il existe un polynôme générateur* $G(X)$*.*

Les polynômes générateurs pour une suite récurrente linéaire donnée $\underline{a}$ forment un idéal de $\mathbb{K}[X]$ dont le générateur unitaire $P = \mathrm{P}^{\underline{a}}$ est appelé le *polynôme générateur minimal* de la suite $\underline{a}$.

Si on note d le degré de $\mathrm{P}^{\underline{a}}$ $(d \leq n)$ et $\mathbf{S}(X) = \sum_{i=0}^{\infty} a_i X^i$, on obtient dans l'anneau des séries formelles $\mathbb{K}[[X]]$ les deux égalités :

$$\begin{aligned} &\mathbf{S}(X)\,\widehat{G}(X) = F(X) \text{ avec} \quad F \in \mathbb{K}[X] \text{ et } \deg F < n\,. \\ &\mathbf{S}(X)\,\widehat{P}(X) = N(X) \text{ avec} \quad N \in \mathbb{K}[X] \text{ et } \deg N < d\,. \end{aligned}$$

ou encore,

$$\mathbf{S}(X) = \frac{F(X)}{\widehat{G}(X)} = \frac{N(X)}{\widehat{P}(X)} \qquad \text{avec } \deg N < d := \deg P\,.$$

Pour calculer $\mathrm{P}^{\underline{a}}$ à partir des $2n$ premiers termes de la suite récurrente linéaire (sachant qu'il existe un polynôme générateur de degré n) l'idée directrice est qu'on peut calculer le développement en fraction continue de $1/\mathbf{S}(X)$ dans $\mathbb{K}[X]$. En effet ce développemnt est le même que celui de $\frac{\widehat{P}(X)}{N(X)}$ et s'arrête au bout d'un nombre fini d'étapes :

$$\frac{1}{\mathbf{S}(X)} = \frac{\widehat{P}(X)}{N(X)} = q_1(X) + \cfrac{1}{q_2(X) + \cfrac{1}{q_3(X) + \cdots}}\,. \tag{3.2}$$

Mais on peut faire les calculs seulement modulo X^{2n}, à condition de bien contrôler le moment où on s'arrête.

3.1.2 L'algorithme de Berlekamp-Massey

Rappelons brièvement l'algorithme de Berlekamp-Massey.

On donne dans un corps $\mathbb{K}$ les $2n$ premiers éléments d'une suite récurrente linéaire $\underline{a} = (a_k)_{k\in\mathbb{N}}$ pour laquelle on sait qu'il existe un polynôme générateur de degré n. Le problème est de calculer son polynôme générateur minimal $\mathrm{P}^{\underline{a}}$.

Une telle solution est donnée par l'algorithme de Berlekamp-Massey qui donne en sortie le degré d ainsi que les coefficients d'un polynôme $f = c_d\,\mathrm{P}^{\underline{a}}$ associé au polynôme $\mathrm{P}^{\underline{a}}$. Ce polynôme $\mathrm{P}^{\underline{a}}$ est alors obtenu en divisant f par c_d.

Cet algorithme a été inventé par Berlekamp en 1968 [6] dans le but de décoder les codes BCH [21], mais sous une forme où la relation avec l'algorithme d'Euclide étendu était invisible. L'algorithme a été « expliqué » une année plus tard par Massey [28] et Dornstetter [14] qui ont montré qu'on pouvait le voir comme une variante de l'algorithme d'Euclide étendu. Voici ce que cela donne.

L'algorithme utilise les propriétés de la suite des triplets (R_i, U_i, V_i) formée des restes et des multiplicateurs de Bézout successifs dans l'algorithme d'Euclide étendu pour le couple de polynômes (R_{-1}, R_0) où $R_{-1} = X^{2n}$ et $R_0 = \sum_{k=0}^{2n-1} a_k X^k$.

Posant $V_{-1} = U_0 = 0$ et $U_{-1} = V_0 = 1$, ces triplets vérifient, pour tout $i \geq 0$, les relations :

$$\begin{aligned}
R_{i-1} &= R_i\,Q_i + R_{i+1} \qquad \text{où } \deg R_{i+1} < \deg R_i \\
U_{i+1} &= U_{i-1} - Q_i\,U_i, \\
V_{i+1} &= V_{i-1} - Q_i\,V_i, \\
\text{d'où :}\quad R_i &= U_i\,R_{-1} + V_i\,R_0\,. \\
\text{De plus :}\quad & U_i\,V_{i-1} - V_i\,U_{i-1} = (-1)^{i+1} \\
\text{et :}\quad & \deg R_i < 2n - \deg V_i\,.
\end{aligned}$$

Les deux dernières relations se vérifient facilement par récurrence sur i.

On arrête le processus au premier reste, disons R_m, de degré plus bas que n, pour obtenir :

$$U_m\,X^{2n} + V_m\,R_0 = R_m \quad \text{avec} \quad \deg R_m < n.$$

Posons $d = \sup(\deg V_m, 1 + \deg R_m)$ et $P = X^d V_m(1/X)$. Alors on peut montrer que P divisé par son coefficient dominant est le polynôme générateur minimal de la suite (a_k) (voir [19] et [14]). Par exemple dans le cas où $\deg V_m = n$ et $V_m(0) \neq 0$, en écrivant que les termes de degré

compris entre n et $2n-1$ du polynôme $V_m(X)\,R_0(X)$ sont nuls, on constate que $P(X)$ est bien un polynôme générateur de la suite (a_k).

Ceci donne précisément l'algorithme 3.1 (dans lequel $\mathrm{cd}(P)$ désigne le coefficient dominant de P).

Algorithme 3.1. ***Algorithme de Berlekamp-Massey usuel***

Entrée : Un entier $n \geq 1$. Une liste non nulle d'éléments du corps $\mathbb{K}$, $[a_0, a_1, \ldots, a_{2n-1}]$: les $2n$ premiers termes d'une suite récurrente linéaire $\underline{a}$, sous l'hypothèse qu'elle admet un polynôme générateur de degré $\leq n$.

Sortie : Le polynôme générateur minimal $\mathrm{P}^{\underline{a}}$ de la suite récurrente linéaire.

Début

Variables locales : $R, R_0, R_1, V, V_0, V_1, Q, P$: polynômes en X

initialisation

$R_0 := X^{2n}$; $R_1 := \sum_{i=0}^{2n-1} a_i X^i$; $V_0 = 0$; $V_1 = 1$;

boucle

tant que $\deg(R_1) \geq n$ **faire**

$(Q, R) :=$ quotient et reste de la division de R_0 par R_1 ;

$V := V_0 - Q\,V_1$;

$V_0 := V_1$; $V_1 := V$; $R_0 := R_1$; $R_1 := R$;

fin tant que

sortie de la boucle

$d := \sup(\deg(V_1), 1 + \deg(R_1))$; $P := X^d V_1(1/X)$; Retourner $\mathrm{P}^{\underline{a}} = P/\mathrm{cd}(P)$.

Fin.

Bien que très simple cet algorithme a toujours semblé un peu trop difficile à justifier. Une littérature considérable a été développée à son sujet, citons par exemple [10, 14, 25, 30, 31, 33, 34].

Nous proposons ici d'expliquer cet algorithme de manière vraiment simple et convaincante, mais pour cela nous avons besoin d'introduire une légère variation, très naturelle, dont, de manière étrange, nous n'avons pas trouvé trace dans la littérature.

Notons que cette explication et cette variante ont été publiées en anglais dans l'article [5]. Signalons aussi qu'après la soumission de cet article est paru le livre de Shoup [32] dans lequel cette variante apparaît sans aucune référence.

Complexité de l'algorithme de Berlekamp-Massey

Nous commençons par le calcul de la suite des quotients. Le nombre d'opérations arithmétiques est maximum lorsque les degrés des restes successifs diminuent de 1 en 1. Cela donne le calcul suivant :

- À l'étape 1 on fait 1 division, $4n-1$ multiplications et $2n-2$ soustractions.
- À l'étape k $(2 \leq k \leq n)$ on fait 1 division, $2(2n-k+1)$ multiplications et $2(2n-k)$ soustractions/additions.
- À la sortie de la boucle on a fait n divisions.

En tout cela fait $2n$ divisions, $3n^2+n-1$ multiplications et $3n^2-3n$ soustractions/additions.

Concernant l'évaluation successive des V à partir de la suite des restes, on obtient

- À l'étape 1 la complexité est nulle (multiplication de deux coefficients par -1).
- À l'étape k $(2 \leq k \leq n)$ on fait $2k$ multiplications et $2(k-1)$ soustractions/additions.

Cela fait en tout $n(n+1)-2$ multiplications et $n(n-1)$ soustractions/additions.

Proposition 3.1. *Le nombre d'opérations arithmétiques, pour le calcul des quotients et des restes, lorsqu'on exécute l'algorithme de Berlekamp-Massey, est majoré par* $6n^2-1$. *Le calcul de* P *à partir de la suite des quotients coûte* $2n^2-2$ *opérations arithmétiques.*

3.2 La variante et sa justification

La variation que nous introduisons est basée sur l'idée suivante. Puisqu'à la fin de l'algorithme, le polynôme V doit être renversé selon une procédure difficile à expliquer (pourquoi doit-on prendre le polynôme réciproque prècisément en degré $d = \sup(\deg(V_1), 1+\deg(R_1))$?), le mieux ne serait-il pas de traiter directement la suite récurrente linéaire « dans le bon sens » (elle a elle-même été renversée au départ lorsqu'on a affecté $R_1 := \sum_{i=0}^{2n-1} a_i X^i$) ?

Naturellement l'appréciation selon laquelle la suite récurrente linéaire a été renversée au départ lorsqu'on a écrit $R_1 := \sum_{i=0}^{2n-1} a_i X^i$ est assez subjective. Elle est en fait renforcée par la remarque suivante. Si le polynôme générateur minimal est de degré d nettement plus petit que n les calculs dans l'algorithme ne devraient pas être sensiblement différents lorsqu'on travaille avec les $2d$ premiers termes de la suite ou lorsqu'on travaille avec les $2n$ premiers termes. Or le renversement effectué au début de l'algorithme change complètement le calcul qui est fait. Tandis qu'en l'absence de renversement, avec notre variante, le calcul sur la suite courte peut facilement être regardé comme le calcul sur la suite longue, tronqué de manière convenable.

Une autre confirmation est le caractère plus simple, et plus facile à justifier, de l'affectation finale.

3.2.1 Une variante de l'algorithme de Berlekamp-Massey : Algorithme 3.2

Algorithme 3.2. ***Algorithme de Berlekamp-Massey, variante***

Entrée : n entier $n \geq 1$. Une liste non nulle d'éléments du corps $\mathbb{K}$, $[a_0, a_1, \ldots, a_{2n-1}]$: les $2n$ premiers termes d'une suite récurrente linéaire $\underline{a}$, sous l'hypothèse qu'elle admet un polynôme générateur de degré $\leq n$.

Sortie : Le polynôme générateur minimal $\mathrm{P}^{\underline{a}}$ de la suite récurrente linéaire.

Début

Variables locales : $R, R_0, R_1, V, V_0, V_1, Q$: polynômes en X ; n' : entier.

initialisation

$n' := 2n-1$; $R_0 := X^{2n}$; $R_1 := \sum_{i=0}^{n'} a_{n'-i} X^i$; $V_0 = 0$; $V_1 = 1$;

boucle

tant que $\deg(R_1) \geq n$ **faire**

$(Q, R) :=$ quotient et reste de la division de R_0 par R_1 ;

$V := V_0 - Q\,V_1$;

$V_0 := V_1$; $V_1 := V$; $R_0 := R_1$; $R_1 := R$;

fin tant que

sortie de la boucle

Retourner $\mathrm{P}^{\underline{a}} := V_1/\mathrm{cd}(V_1)$.

Fin.

3.2.2 Preuve de correction de l'algorithme 3.2

La preuve de correction de l'algorithme 3.2, nécessite les définitions et les résultats suivants. Si $\underline{a} = (a_n)_{n\in\mathbb{N}}$ est une suite (infinie) arbitraire et si i, r, $p \in \mathbb{N}$ nous noterons :

$$\mathrm{H}^{\underline{a}}_{i,r,p} = \mathrm{Hk}(a_i, a_{i+1}, \ldots, a_{i+r+p-2}; r; p) \begin{pmatrix} a_i & a_{i+1} & a_{i+2} & \cdots & a_{i+p-1} \\ a_{i+1} & a_{i+2} & & & a_{i+p} \\ a_{i+2} & & & & \\ \vdots & & & & \vdots \\ a_{i+r-1} & a_{i+r} & \cdots & \cdots & a_{i+r+p-2} \end{pmatrix}.$$

La proposition suivante est classique (voir par exemple [1]).

Proposition 3.2. *Soit $\underline{a}$ une suite récurrente linéaire qui admet un polynôme générateur de degré $\leq n$. On a alors : $d = \deg(\mathrm{P}^{\underline{a}}) = \mathrm{rang}(\mathrm{H}^{\underline{a}}_{0,n,n})$ où*

$$\mathrm{H}^{\underline{a}}_{0,n,n} \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ a_1 & a_2 & & \iddots & a_n \\ a_2 & & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & & \vdots \\ a_{n-1} & a_n & \cdots & \cdots & a_{2n-2} \end{pmatrix}.$$

Les coefficients de $\mathrm{P}^{\underline{a}}(X) = X^d - \sum_{i=0}^{d-1} g_i X^i \in \mathbb{K}[X]$ sont l'unique solution de l'équation

$$\mathrm{H}^{\underline{a}}_{0,d,d}\, G = \mathrm{H}^{\underline{a}}_{d,d,1}$$

c'est-à-dire encore l'unique solution du système linéaire

$$\begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{d-1} \\ a_1 & a_2 & & \iddots & a_d \\ a_2 & & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & & \vdots \\ a_{d-1} & a_d & \cdots & \cdots & a_{2d-2} \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{d-1} \end{pmatrix} = \begin{pmatrix} a_d \\ a_{d+1} \\ a_{d+2} \\ \vdots \\ a_{2d-1} \end{pmatrix}. \tag{3.3}$$

On en déduit facilement la précision suivante.

Proposition 3.3. *En outre pour qu'un vecteur $Y = (p_0, \ldots, p_n)$ soit solution de l'équation*

$$\mathrm{H}^{\underline{a}}_{0,n,n+1}\, Y = 0$$

c'est-à-dire

$$\begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} & a_n \\ a_1 & a_2 & & \iddots & a_n & a_{n+1} \\ a_2 & & \iddots & \iddots & \vdots & \vdots \\ \vdots & \iddots & \iddots & & \vdots & \vdots \\ a_{n-1} & a_n & \cdots & \cdots & a_{2n-2} & a_{2n-1} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{pmatrix} \tag{3.4}$$

il faut et il suffit que le polynôme $P(X) = \sum_{i=0}^{n} p_i X^i \in \mathbb{K}[X]$ soit multiple de $\mathrm{P}^{\underline{a}}(X)$.

Démonstration. Puisque la matrice est de rang d son noyau est de rang $n+1-d$. Puisque la suite est une suite récurrente linéaire ayant $\mathrm{P}^{\underline{a}}$ pour polynôme générateur, les polynômes $\mathrm{P}^{\underline{a}}$, $X\mathrm{P}^{\underline{a}}$, …, $X^{n-d}\mathrm{P}^{\underline{a}}$ correspondent à des vecteurs du noyau linéairement indépendants. Donc tout élément du noyau est une combinaison linéaire de ces vecteurs colonnes. □

Par ailleurs, nous faisons la constatation suivante.

Fait 3.2.1. *En posant* $n' := 2n-1$ *et* $\widehat{S}(X) := \sum_{i=0}^{n'} a_{n'-i}X^i$, *l'équation (3.4) est équivalente à l'affirmation suivante :*

- *Le polynôme* P *est de degré* $\leq n$ *et on a :*

$$\exists R \in \mathbb{K}[X] \text{ tel que } \quad \deg(R) < n \quad \text{et} \quad P(X)\widehat{S}(X) \equiv R(X) \mod X^{2n}.$$

Autrement dit donner une solution de (3.4) revient à trouver des polynômes P, R, U *tels que :*

$$\deg(R) < n,\ \deg(P) \leq n \quad \text{et} \quad P(X)\widehat{S}(X) + U(X)X^{2n} = R(X) \tag{3.5}$$

Le problème de trouver le polynôme générateur minimal de $\underline{a}$ est donc ramené au problème de réaliser les conditions (3.5) avec le degré de P minimum.

Nous avons par ailleurs le fait suivant « bien connu » concernant l'algorithme d'Euclide étendu.

Fait 3.2.2. *Soient* R_0 *et* R_1 *de degrés* p *et* $q \leq p$, *et un entier* $n < q$. *Supposons que le degré du pgcd de* R_0 *et* R_1 *est* $< n$. *Notons* $R_0, R_1, \ldots, R_s$ *la suite des restes dans l'algorithme d'Euclide.*

1. *L'algorithme d'Euclide étendu démarrant avec* R_0 *et* R_1 *réalise des équations (du même type que (3.5))*
$$V_k(X)R_1(X) + U_k(X)R_0 = R_k(X)$$
Lorsque le premier reste R_k *de degré* $< n$ *est atteint, on a* $\deg(V_k) \leq p-n$ *et* $\deg(U_k) \leq q-n$.
2. *Il n'est pas possible d'obtenir une équation du même type*
$$V(X)R_1(X) + U(X)R_0 = R(X)$$
avec $\deg(R) < \deg(R_{k-1})$ *et* $\deg(V) < \deg(V_k)$.

Démonstration. Pour ℓ fixé on considère l'espace vectoriel E_ℓ formé par les polynômes $V(X)R_1(X) + U(X)R_0$ avec $\deg(U) \leq \ell$ et $\deg(V) \leq \ell + p - q$. On a $E_{\ell+1} = E_\ell + XE_\ell$, avec $1 + \dim(E_\ell) \leq \dim(E_{\ell+1}) \leq 2 + \dim(E_\ell)$. Disons que « le degré k est présent dans E_ℓ » s'il y a dans E_ℓ un polynôme de degré k. Notons Δ_ℓ l'ensemble des degrés présents dans E_ℓ. Le cardinal de Δ_ℓ est égal à la dimension de E_ℓ. On a aussi $\Delta_\ell \cup (1+\Delta_\ell) \subseteq \Delta_{\ell+1}$, avec égalité si le cardinal augmente de 2 entre Δ_ℓ et $\Delta_\ell \cup (1+\Delta_\ell)$. Comme $\#(\Delta_{\ell+1}) \leq 2 + \#(\Delta_\ell)$, l'ensemble Δ_ℓ rangé en ordre croissant ne peut pas contenir plus qu'un trou.
Prenons un exemple (voir le tableau 3.1 page suivante). Supposons que $p = 12$, $q = 10$ et que les degrés dans la suite des restes soient 7, 6, 2, 1 (avec le pgcd de degré 1). Alors les relations précédemment établies entre $\Delta_{\ell+1}$ et Δ_ℓ impliquent que les Δ_ℓ successifs sont les suivants (on a rendu le trou éventuel visible par un blanc) :

$\Delta_0 =$	7, 10, 11, 12,
$\Delta_1 =$	7, 8, 10, 11, 12, 13,
$\Delta_2 =$	7, 8, 9, 10, 11, 12, 13, 14,
$\Delta_3 =$	6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
$\Delta_4 =$	2, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
$\Delta_5 =$	2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
$\Delta_6 =$	2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
$\Delta_7 =$	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
$\Delta_8 =$	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
$\Delta_9 =$	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
	etc . . .

Tableau 3.1 – Degrés présents dans les E_ℓ successifs

Notons δ_k le plus petit élément de Δ_k. On voit alors que tout δ_k est un degré dans la suite des restes. En outre si le degré du pgcd est $< q-k$, on a toujours $\delta_k < q-k$. Enfin si le degré ℓ est atteint la première fois avec δ_k, on a $k \leq q-\ell$, et aucun degré $< \delta_{k-1}$ n'est atteint par un E_h où $h < k$. Ceci prouve les affirmations 1 et 2. □

En appliquant le fait 3.2.2 avec $R_0 = X^{2n}$, $n' = 2n-1$, $R_1 = \widehat{S} = \sum_{i=0}^{n'} a_{n'-i}X^i$ et $q = \deg R_1$ on obtient donc :

- l'algorithme d'Euclide étendu démarrant avec $R_0 = X^{2n}$ et $R_1 = \widehat{S}$ réalise une équation telle que (3.5) avec $R = R_k$ lorsque le premier reste R_k de degré $< n$ est atteint,
- quand une équation telle que (3.5) est réalisée par l'algorithme d'Euclide étendu, il n'est pas possible d'obtenir une équation du même type $P'(X)\widehat{S}(X) + U'(X)X^{2n}R'(X)$ avec $\deg(R') < \deg(R_{k-1})$ et $\deg(P') < \deg(P)$.

Ceci termine la preuve de correction de l'algorithme 3.2.

3.3 Application des algorithmes de diagonalisation aux suites récurrentes linéaires

L'algorithme 3.2 a manifestement la même complexité que l'algorithme de Berlekamp-Massey usuel. Cependant nous avons vu dans les chapitres précédents le lien très étroit entre l'algorithme de diagonalisation d'une matrice de Hankel et l'algorithme d'Euclide étendu, interrompu en son milieu, qui est utilisé dans l'algorithme 3.2, et nous avons remarqué qu'on pouvait améliorer légèrement ses performances en ne calculant que la partie utile des restes successifs. Ceci débouche sur deux variantes 3.3 et 3.4 des algorithmes de diagonalisation 1.2 et 1.3, qui donnent une accélération de l'algorithme 3.2.

3.3.1 Algorithme 3.3

Avant de donner l'algorithme 3.3, nous rappelons les résultats du théoreme 9.17 (page 334) du livre [3], qui relient les suites récurrentes linéaires et les matrices de Hankel.

Théorème 3.1. *Soit $\mathbb{K}$ un corps. Soit $\underline{a} = (a_k)_{k\in\mathbb{N}}$ une suite infinie d'éléments de $\mathbb{K}$ et $n \in \mathbb{N}$. Les conditions suivantes sont équivalentes :*

(*a*) *La suite* $\underline{a}$ *vérifie une relation récurrente linéaire d'ordre* n *à coefficients dans* $\mathbb{K}$ *:*

$$c_n a_p = -c_{n-1}a_{p-1} - \cdots - c_0 a_{p-n},$$

où $c_n \neq 0$ *et* $p \geq n$.

(*b*) *Il existe un polynôme* $P \in \mathbb{K}[X]$ *de degré* n *et une forme linéaire* λ *sur* $\mathbb{K}[X]/(P)$ *tel que* $\lambda(X^i) = a_i$ *pour tout* $i \geq 0$.

(*c*) *Il existe des polynômes* $P, Q \in \mathbb{K}[X]$ *avec* $\deg(Q) < \deg(P) = n$, *tel que :*

$$\frac{Q(X)}{P(X)} = \sum_{i=0}^{\infty} \frac{a_i}{X^{i+1}}$$

(*d*) *Il existe* $r \leq n$ *pour lequel on a :*

$$r \;=\; \operatorname{rang}(\mathrm{H}^{\underline{a}}_{0,r,r}) = \operatorname{rang}(\mathrm{H}^{\underline{a}}_{0,r+1,r+1}) = \operatorname{rang}(\mathrm{H}^{\underline{a}}_{0,r+2,r+2}) = \cdots$$

(*e*) *Il existe* $r \leq n$ *pour lequel on a :*

- $\det(\mathrm{H}^{\underline{a}}_{0,r,r}) \neq 0$,
- $\det(\mathrm{H}^{\underline{a}}_{0,p,p}) = 0$; $p > r$.

Vues la proposition 1.2 et le lemme 1.4.1, l'algorithme 1.2 calcule, dans le cas où le dernier bloc de la réduite est nul, les quotients et restes successifs pour le développement en fraction continue de $1/S(X)$ dans l'anneau des développements limités à l'ordre $2n$. Il permet donc de récupérer le polynôme générateur minimal de la suite récurrente linéaire. Si on part d'une matrice de Hankel arbitraire, elle ne correspondra pas en général à une suite récurrente linéaire parce que sa réduite diagonale ne sera pas de la forme voulue : le dernier bloc de Hankel inférieur ne sera pas identiquement nul.

Maintenant si on considère seulement les $2n$ premiers termes de la suite récurrente linéaire on peut prendre le $(2n+1)^{\text{ème}}$ terme comme une indéterminée (comme dans les exemples 1.3.1 et 1.3.2). Cette indéterminée n'intervient à chaque fois que dans le dernier terme (celui du degré formel dominant) du développement limité. On peut ne pas faire les calculs correspondants, et la structure même du calcul nous montre que le dernier terme en question est toujours du type $\lambda b + \mu$ avec $\lambda \neq 0$ et il pourra toujours être rendu nul par le choix convenable de la valeur de b.

Corollaire 3.3.1. *Soit* $\mathbb{K}$ *un corps et* $\underline{a} = (a_k)_{k\in\mathbb{N}}$ *une suite infinie d'éléments de* $\mathbb{K}$. *Alors,* $\underline{a}$ *est une suite récurrente linéaire d'ordre* n **si et seulement si** *pour tout* $m \geq n$, *la réduite diagonale par blocs* D *de* $\mathrm{H}^{\underline{a}}_{0,m+1,m+1}$) *est de la forme :*

$$D = \begin{pmatrix} (D_{11}) & & & & \\ & (D_{22}) & & & \\ & & \ddots & & \\ & & & (D_{kk}) & \\ & & & & (O_{m-n\times m-n}) \end{pmatrix}.$$

Algorithme 3.3. ***Algorithme de type Berlekamp-Massey, variante améliorée (1) (en fait une variante de l'algorithme 1.2)***

Entrée : Une liste non nulle d'éléments du corps $\mathbb{K}$, $H = [\alpha_1, \alpha_2, \ldots, \alpha_{2n}]$, sauf exception ce sont les $2n$ premiers termes d'une suite récurrente linéaire.
Sortie : Stu : une liste de quotients formels. P : le polynôme générateur minimal.
Variables locales : m, r, p, s : compteurs ; R_0, R_1, R_2, Q : polynômes à coefficients dans $\mathbb{K}$.
Début
  # initialisation
  $Stu := [\,]$; $r :=$ l'indice du premier coefficient non nul de H ; $m := n$; $p := 2n - r$; $R_0 := 1$;
  $R_1 := \alpha_r + \alpha_{r+1}X + \ldots + \alpha_{p+r}X^p := \sum_{k=r}^{p+r} \alpha_k X^{k-r}$;
  # boucle
  tant que $r < p$ **faire**
    $[Q, R_2] :=$QuoResCroiss(R_0, R_1, r, p) ;
    $Stu := Stu \bullet [[Q, r]]$;
    $s :=$ valuation de R_2 ;
    $p := p - s$; $m := m - r$; $r := s - r$;
    $R_0 := R_1 \mod X^{p+1}$;
    $R_1 := -R_2/X^s$;
  fin tant que ;
  # sortie
  si $p < 0$ **alors**
    Calculer le polynôme générateur minimal P à partir de la suite Stu ;
    # cf. formule 1.21
    Retourner $[Stu, P]$
  sinon
    Signaler : les données ne correspondent pas à une suite récurrente linéaire ;
    Retourner $[Stu]$
  fin si
Fin.

Donc dans le tout dernier bloc Hankel-inférieur fourni par la réduction, ou bien tous les termes effectivement calculés sont nuls (c'est par exemple le cas si le dernier bloc de Hankel est d'ordre 1 car seul subsiste le terme non calculé qui peut être rendu nul), et les $2n$ termes de la suite correspondent bien à une suite récurrente linéaire ayant un polynôme générateur minimal de degré $\leq n$, ou bien ce n'est pas le cas, et notre algorithme nous donne l'information correspondante.

Ceci donne l'algorithme 3.3, variante de l'algorithme 1.2.

Complexité de l'algorithme 3.3

Le calcul de la suite Stu des quotients partiels est du même ordre que l'algorithme 1.2 pour une matrice de Hankel d'ordre $n + 1$. Donc le nombre total d'opérations arithmétiques est asymptotiquement de l'ordre de $4n^2$. Puisque le calcul du polynôme minimal à partir de la suite des quotients coûte asymptotiquement $2n^2$, le coût total est asymptotiquement $6n^2$.

3.3.2 Algorithme 3.4

De même que l'algorithme 3.3 est obtenu à partir de l'algorithme 1.2, l'algorithme suivant 3.4 est obtenu à partir de l'algorithme 1.3.

Algorithme 3.4. ***Algorithme de type Berlekamp-Massey, variante améliorée (2) (en fait une variante de l'algorithme 3.3)***

Entrée : Une liste non nulle d'éléments du corps $\mathbb{K}$, $H = [\alpha_1, \alpha_2, \ldots, \alpha_{2n}]$, sauf exception ce sont les $2n$ premiers termes d'une suite récurrente linéaire.
Sortie : Stu : la liste des quotients partiels. P : le polynôme générateur minimal.
Variables locales : $m, r, d_0, d_1, d_2, dd_1, dd2$: compteurs ; $V, V_0, V_1, R_0, R_1, R_2, Q,$: polynômes.
Début

initialisation
$m := n$; $R_0 := X^{2n}$; $R_1 := \sum_{k=0}^{2n-1} \alpha_{k+1} X^{2n-1-k}$; $Stu := [\,]$;
$d_0 := \deg R_0$; $d_1 := \deg R_1$; $V_0 = 0$; $V_1 = 1$;
boucle
tant que $2d_1 - d_0 + 1 > 0$ **faire**
$dd_1 := d_0 - d_1 - 1$;
$Q :=$ le quotient de la division de R_0 par R_1 ;
$V := V_0 - Q\,V_1$; $V_0 := V_1$; $V_1 := V$;
$R_2 :=$ le reste de la division de R_0 par R_1 ;
$R_1 := \mathrm{Tronk}(R_1, X, dd_1)$;
$R_2 := \mathrm{Tronk}(-R_2, X, dd_1)$;
$d_1 := \deg R_1$; $d_2 := \deg R_2$; $dd_2 := d_1 - d_2$;
$R_0 := \mathrm{Tronq}(R_1, X, dd_2)$;
$R_1 := \mathrm{Tronq}(R_2, X, 0)$;
$Stu := Stu \bullet [Q]$;
$r := \deg Q$; $m := m - r$;
$d_0 := \deg R_0$; $d_1 := \deg R_1$;
fin tant que ;
sortie
si $d_1 < 0$ **alors**
$P := V_1/\mathrm{cd}(V_1)$;
Retourner $[Stu, P]$;
sinon
Signaler : les données ne correspondent pas à une suite récurrente linéaire ;
Retourner $[Stu]$;
fin si
Fin.

La sortie de l'algorithme est une suite de quotients, notée Stu, dans un algorithme d'Euclide « tronqué ». Le quotient Q_i code la matrice Hankel-inférieure d'ordre $\deg Q_i$, dont la dernière colonne est donnée par les coefficients de Q_i, rangés par degré décroissants (en omettant le coefficient constant). Le polynôme générateur minimal est récupéré à partir de la suite Stu comme dans l'algorithme 3.3.

On notera que les Q_i dans l'algorithme 3.4 sont les polynômes réciproques de ceux calculés par l'algorithme 3.3. Dans l'algorithme 3.3 il était nécessaire que les quotients soient des polynômes formels tandis que dans l'algorithme 3.4 les polynômes réciproques sont des polynômes

usuels (cf. la formule 1.22). Par ailleurs l'algorithme 3.4 est aussi une variante de l'algorithme 3.2 présenté dans la section 3.2.

Enfin la correction de l'algorithme 3.4 découle immédiatement de celle de l'algorithme 3.3 qui est elle même conséquence de celle de l'algorithme 1.2. Quant à la complexité elle est asymptotiquement $6n^2$: la même que celle de l'algorithme 3.3.

3.3.3 Comparaison entre l'algorithme de Berlekamp-Massey et l'algorithme 3.4

Dans l'algorithme de Berlekamp-Massey 3.1 ou sa variante 3.2 on suppose qu'on a bien en entrée les $2n$ premiers termes d'une suite récurrente linéaire ayant un polynôme générateur minimal de degré $\leq n$. Par ailleurs l'algorithme des quotients et restes est effectué sans troncatures.

Dans l'algorithme 3.3 ou sa variante 3.4 on ne fait aucune hypothèse sur l'entrée. L'algorithme teste si l'entrée est correcte et fournit la suite des quotients, sans jamais effectuer aucun calcul inutile grâce aux troncatures bien contrôlées.

Ces algorithmes sont de complexité asymptotique $6n^2$ et économisent le quart du temps de calcul de l'algorithme de Berlekamp-Massey usuel 3.1 (ou de sa variante 3.2).

3.4 Variante « dynamique » de l'algorithme 3.2

Notre algorithme de Berlekamp-Massey modifié permet une évaluation paresseuse (on dit aussi évaluation dynamique, dans l'esprit de D5 [11]) que nous illustrons sur le problème suivant.

Soit $f(x) \in \mathbb{K}[x]$ un polynôme séparable de degré δ. Soit B l'algèbre de décomposition universelle de f, A une algèbre quotient de B et $a \in A$. Ainsi A est une algèbre zéro-dimensionnelle qu'on décrit sous la forme

$$A \simeq \mathbb{K}[X_1, \ldots, X_\delta] / \langle f_1, \ldots, f_s \rangle ,$$

où $f_1, \ldots, f_s$ définit une base de Gröbner pour un ordre monomial convenable.

Notre but est de calculer le polynôme minimal de a, ou, au moins, l'un de ses facteurs. Cependant la dimension de A comme $\mathbb{K}$-espace vectoriel, notée m, est en général trop grande pour pouvoir manipuler des matrices d'ordre m. On applique donc l'idée de l'algorithme de Wiedeman, qui est de calculer les $2m$ premiers termes de la suite récurrente linéaire $a_k = \phi(a^k)$, où ϕ est une forme linéaire sur A. Le polynôme générateur minimal de cette suite récurrente linéaire est un facteur du polynôme minimal de a. Il est égal au polynôme minimal de a avec une grande probabilité.

Comme le calcul de a^k pour k grand est en général très cher et comme on espère que le polynôme minimal a un degré d nettement plus petit que m, on est intéressé à essayer de faire le calcul avec seulement un nombre $2d < 2m$ de termes de la suite récurrente linéaire.

Le problème est qu'on ne connaît pas d par avance. Par contre on dispose souvent d'une majoration n pour d, avec $n < m$. Néanmoins, on veut éviter autant que possible de calculer les $2n$ premières puissances de a.

En conséquence on choisit un $\ell < n$. On fait tourner l'algorithme 3.2 avec ℓ and $[\phi(a^0), \ldots, \phi(a^{2\ell-1})]$ comme entrée. Si P est le polynôme obtenu (de degré $\leq \ell$), on peut tester si $P(a) = 0$, auquel cas P est le polynôme minimal recherché.

Sinon on choisit un ℓ', $\ell < \ell' = \ell + r \leq n$, et on répète le calcul.

Algorithme 3.5. ***Algorithme de Berlekamp-Massey paresseux (dans un contexte particulier)***

Entrée : $n, r, l \in \mathbb{N}$, G : base de Gröbner, $a \in A$, phi : $A \to \mathbb{K}$ programme qui évalue $\phi(y)$.
Le polynôme minimal de a a son degré majoré par n.
Sortie : Le polynôme minimal P de a, ou, à défaut, un facteur de ce polynôme minimal.
Début
Variables locales : m, l, i : entiers, $R, R_{-1}, R_0, R_1, V, V_{-1}, V_0, V_1, U, U_{-1}, U_0, U_1, S_0, S_1, Q$: polynômes dans $\mathbb{K}[x]$, L : liste dans A, W : liste dans $\mathbb{K}$, $val \in A$;
initialisation
$L := [1, a]$; $W := [\mathsf{phi}(1), \mathsf{phi}(a)]$; $S_0 := x^{2l}$; $S_1 = W[1]\, x^{2l-1} + W[2]\, x^{2l-2}$;
boucle
pour i **de** 3 **à** $2l$ **faire**
$L[i] := \mathsf{normalf}(L[i-1] \times a, G)$; $W[i] := \mathsf{phi}(L[i])$; $S_1 = S_1 + W[i]x^{2l-i}$;
fin pour ;
$R_0 := S_0$; $R_1 := S_1$; $V_0 = 0$; $V_1 = 1$; $U_0 = 1$; $V_1 = 0$;
boucle
tant que $l \leq \deg(R_1)$ **faire**
$(Q, R) :=$ quotient et reste de la division de R_0 par R_1 ;
$V := V_0 - QV_1$; $U := U_0 - QU_1$; $U_{-1} := U_0$; $V_{-1} := V_0$;
$V_0 := V_1$; $V_1 := V$; $U_0 := U_1$; $U_1 := U$; $R_0 := R_1$; $R_1 := R$;
fin tant que ;
$val := \mathsf{Subs}(x = a, V_1)$;
boucle
tant que $val \neq 0$ et $l < n$ **faire**
$l := l + r$;
boucle
pour i **de** $2(l-r)+1$ **à** $2 \cdot \inf(l, n)$ **faire**
$L[i] := \mathsf{normalf}(L[i-1] \times a, G)$; $W[i] := \mathsf{phi}(L[i])$;
fin pour ;
$S_0 = x^2 S_0$; $S_1 = x^2 S_1 + W[2l-1]x + W[2l]$;
$R_0 := U_{-1}S_0 + V_{-1}S_1$; $R_1 := U_0 S_0 + V_0 S_1$;
$U1 := U_0; V_1 := V_0; U_0 := U_{-1}; V_0 := V_{-1}$;
boucle
tant que $\inf(l, n) \leq \deg(R_1)$ **faire**
$(Q, R) :=$ quotient et reste de la division de R_0 par R_1 ;
$V := V_0 - QV_1$; $U := U_0 - QU_1$; $U_{-1} := U_0$; $V_{-1} := V_0$;
$V_0 := V_1$; $V_1 := V$;$U_0 := U_1$; $U_1 := U$; $R_0 := R_1$; $R_1 := R$;
fin tant que ;
$val := \mathsf{Subs}(x = a, V_1)$;
fin tant que ; # sortie
Return $P := V_1/\mathrm{cd}(P)$.
Fin.

Dans cette nouvelle étape, il est possible de tirer avantage du calcul des quotients partiels effectué dans la première étape (à l'exception du dernier) de sorte que l'algorithme d'Euclide étendu démarre avec $R_0 = U_0 x^{2\ell'} + V_0 \sum_{i=0}^{2\ell'-1} (\phi(x^{2\ell'-1-i})x^i)$ et $R_1 = U_1 x^{2\ell'} + V_1 \sum_{i=0}^{2\ell'-1} (\phi(x^{2\ell'-1-i})x^i)$,

où U_0, V_0, U_1 et V_1 sont les coefficients de Bezout calculés à la première étape.

L'algorithme 3.5 constitue une première tentative, qui a pu donner des essais concluants. Naturellement le choix de ℓ et r n'est pas unique. Nous proposons $\ell = n/4$, $r = \sup(2, n/16)$. En pratique les caractéristiques particulières du problème considéré peuvent aider à choisir un ℓ et un r convenables.

Enfin, la simplification de l'algorithme d'Euclide présentée dans [14] peut être prise en compte pour optimiser la procédure.

Conclusion

La thèse est consacrée à montrer comment traiter de façon efficace les matrices de Hankel, type particulier et très important des matrices structurées.

Dans cette thèse nous avons montré que plusieurs algorithmes sont sérieusement améliorés. Ceci est prouvé théoriquement et aussi validé par la mise en oeuvre expérimentale.

Plus précisément, la thèse contient :

1. Un nouveau algorithme, simple et rapide, de diagonalisation par blocs Hankel-inférieurs d'une Hankel.
2. Une nouvelle preuve algorithmique, du Théorème de Frobenius concernant la signature d'une matrice de Hankel au moyen des signes de ces mineurs principaux dominants.
3. Une nouvelle version de l'algorithme de Berlekamp-Massey qui sert à calculer le polynôme minimal d'une suite récurrente linéaire avec des éléments dans un corps arbitraire.

Nous concluons alors que la richesse de la diagonalisation par blocs des matrices structurées tels que les matrices de Hankel et l'aspect intuitive que possède cette méthode en calcul formel, nous encourage vivement à persévérer dans cette voie de recherche. Nous pouvons dégager diverses perspectives, dont les plus immédiates sont :

- donner une version du type « algorithme des sous-résultants » pour notre algorithme, ce qui nous permettra de mieux controler la taille des objets intermédiaires lorsqu'on se situe sur un corps infini,
- aussi élucider quelles réductions de la matrice peuvent être obtenues à partir de cette variante,
- calculer précisément les complexités des diverses variantes, tant du point de vue arithmétique que du point de vue binaire,
- obtenir une diagonalisation par blocs d'une Hankel, dans laquelle les blocs diagonaux seront de la forme $\pm J$,
- est-ce que la diagonalisation pré-citée est unique ?
- y a t-il un algorithme facile pour la calculer ?

"Celui qui se perd dans sa passion a moins perdu que celui qui perd sa passion"

Annexe A. Codes Maple

Introduction

La présente annexe a pour objet de rassembler les procédures (écritent en MAPLE), utilisées dans les programmes exécutables qui mettent en évidence les algorithmes présentés et étudiés dans cet ouvrage, quelques unes d'elles étant suivie d'une brève description.

A.1 Les procédures communes aux autres programmes

CB := `proc(A,P,p) ;`
```
  if nargs=3 then EVM(transpose(P)*A*P,p)
  else EVM(transpose(P)*A*P) fi ;
end proc ;
```
description "Changement de base pour la forme quadratique A via la matrice P éventuellement modulo p"

CoeDiag := `proc(A)`
```
local i,ta ;
  ta:= rowdim(A) ;
  [seq(A[i,i], i=1..ta)] ;
end proc ;
```
description "Les coefficients diagonaux de la matrice sont ranges dans une liste"

CoeHK := `proc(A)`
```
local i,n ;
  n:= rowdim(A) ;
  [seq(A[1,i], i=1..n), seq(A[n,i], i=2..n)] ;
end proc ;
```
description "Liste des coefficients de la matrice de Hankel"

CoeTo := `proc(A)`
```
local i,n ;
  n:= rowdim(A) ;
  [seq(A[n-i+1,1], i=1..n), seq(A[1,i], i=2..n)] ;
end proc ;
```
description "Liste des coefficients d'une matrice Toeplitz carrée"

CoeToS := `proc(A)`
```
local i,n ;
  n:= rowdim(A) ;
  [seq(A[1,i], i=1..n)] ;
end proc ;
```
description "Liste des coefficients intéressants d'une matrice Toeplitz supérieure carrée"

DebutListe := proc (L,n)
local i ;
  [seq(L[i],i=1..n)] ;
end proc ;

EVL := proc(L,p)
local j,n,L1 ;
  n := nops(L) ; L1 := copy(L) ;
  if nargs = 2 and p <> 0 then
    for j to n do L1[j] := simplify(normal(L1[j]) mod p) od
  else
    for j to n do L1[j] := normal(L1[j]) od
  fi ;
  [seq(L1[j],j = 1..n)]
end proc ;

EVM := proc(A,p :: nonnegint)
local i,j,nl,nc,B ;
  B := evalm(A) ;
  nl := rowdim(A) ; nc := coldim(A) ;
  if nargs = 2 and p <> 0 then
    for i to nl do
      for j to nc do
        B[i,j] := simplify(normal(B[i,j]) mod p)
      od ;
    od ;
  else
    for i to nl do
      for j to nc do
        B[i,j] := normal(B[i,j])
      od ;
    od ;
  fi ;
  matrix(nl,nc,[seq([seq(B[i,j], j = 1..nc)], i = 1..nl)])
end proc ;

description "Evalue une matrice de polynômes ou fractions rationnelles, éventuellement modulo p"

EVMS := proc(A,p :: nonnegint)
local i,j,nl,nc,p1,B ;
  nl := rowdim(A) ; nc := coldim(A) ;
  if nargs = 1 then p1 := 0 else p1 := p fi ;
  B := EVM(A,p1) ;
  matrix(nl,nc,[seq([seq(sort(expand(B[i,j])), j = 1..nc)], i = 1..nl)]) ;
end proc ;

description "Comme la précédente mais avec les entrees de la matrice triées par le sort de Maple"

InvToS := proc(S,p)
```
local B, A, X, Q, i, n ;
  B := LiPo2(S,X) ; n := nops(S)-1 ; A := X^(2*n) ;
  if nargs = 1 or p = 0 then
    Q := evala(Quo(A,B,X))
  else Q := Quo(A,B,X) mod p fi ;
  [seq(coeff(Q,X,n-i),i = 0..n)] ;
end proc ;
```
description "Inversion d'une Toeplitz supérieure, on ne voit que la liste des coefficients utiles, le calcul utilise une division de polynomes, nettement plus rapide qu'une inversion de matrice generale"

IsDiag := proc(A)
```
local n,i,j,b ;
  n := rowdim(A) ; b := true ;
  for i to n while b do
    for j to n do
      if i <> j then b := evalb(normal(A[i,j]) = 0) fi
    od ;
  od ;
  [b] ;
end proc ;
```
description "La matrice (carrée) est elle diagonale ? "

IsEqm := proc(A,B)
```
local i,j,b ; b := true ;
  for i to rowdim(A) while b do
    for j to coldim(A) while b do
      b := evalb(normal(A[i,j]-B[i,j]) = 0)
    od ;
  od ;
  [b] ;
end proc ;
```
description "Teste si les deux matrices sont égales "

IsHank := proc(A,n)
```
local ish,p,q,k,k0,xk,j ;
  ish := true ; k0 := NULL ;
  for k from 3 to 2*n-1 do
    p := min(n,k-1) ; q := k-p ; xk := A[p,q] ;
    for j from p-1 to max(1,q) by -1 do
      ish := ish and (normal(xk-A[j,k-j]) = 0) ;
      if not (normal(xk-A[j,k-j]) = 0) then k0 := k0,k fi
    od
  od ;
  [ish,k0] ; ;
end proc ;
```
description "Teste si la matrice A est de Hankel.
Si la matrice n'est pas de Hankel, cela donne la liste des endroits ou cela ne va pas"

```
IsHanK := proc(A)
local n ;
  n := rowdim(A) ;
  IsHank(A,n) ;
end proc ;
```
description "La matrice (carrée) est elle Hankel ?"

```
IsTop := proc(A)
local i,n,B ;
  n := rowdim(A) ;
  B := EVM(Ara(n)*A) ;
  IsHanK(B) ;
end proc ;
```
description "Teste si une matrice (supposée carrée) est Toeplitz"

```
LiPo := proc(L,X::name) local i, n ;
n := nops(L) ;
  sum(L[n-i+1]*X^(n-i),i=1..n) ;
end proc ;
```
description "Transformation d'une liste en un polynôme"

```
LiPo2 := proc(L,X::name)
local i, n ;
  n := nops(L) ;
  sum(L[n-i]*X^(i),i=0..n-1) ;
end proc ;
```
description "Transformation d'une liste en un polynôme, dans l'autre sens"

```
LisCoef := proc(P,X::name)
local i,p ;
  p := degree(P,X) ;
  [seq(coeff(P,X,p-i),i=0..p)] ;
end proc ;
```
description "Donne la liste des coeffs d'un polynôme par degrés décroissants, il faut indiquer la variable"

```
LisCoeSer := proc(S,X::name) local P ;
  P := convert(S,polynom) ;
  LisCoef(P,X) ;
end proc ;
```
description "Donne la liste des coefficients de la "série"
préalablement convertie en polynôme "

```
ProToS := proc (S1,S2,p) ;
  if nargs=3 and p<>0 then EVM(LiMa(S1)*VoirToS(S2,p),p)
  else EVM(LiMa(S1)*VoirToS(S2)) fi ;
end proc ;
```
description "Produit de deux Toplitz supérieures, on ne voit que la liste des coefficients.
On a fait le produit de la première ligne de la première Toeptlitz par la seconde Toeplitz,
cela pourrait être accéléré à la Karatsuba ou à la FFT"

QuoRem := proc(A,B,X ::name,p1)
local Q,R,b ;
  b := evalb(nargs=4 and p1 <> 0) ;
  R := 'R' ;
  if b then Q := Quo(A,B,X,'R') mod p1
  else Q := evala(Quo(A,B,X,'R'))
  [Q,R] ;
end proc ;
  description "On retourne le quotient et le reste (éventuellement modulo p) de la division de A par B"

ReDL := proc(A,X::name,q) ;
  sum(coeff(A,X,i)*X^i,i=0..q) ;
end proc ;
  description "Développement limité à l'ordre q du polynôme A en X"

SumDL := proc(A,B,X::name,q,p)
local M ;
  if nargs=4 or p=0 then
    M := expand(A+B) ;
  else M := expand(A+B) mod p ; fi ;
  ReDL(M,X,q) ;
end proc ;
  description "Somme de deux développements limités"

Taille := proc(H)
local n ;
  n := nops(H) ;
  if irem(n,2) <> 1 then
    print(` erreur dans Taille, la liste a un nb pair d'elts `)
  else (n+1)/2 fi ;
end proc ;
  description "Taille d'une Hankel codée par une liste"

ValDL := proc(P,X::name,q)
local j ;
  j := 0 ;
  while coeff(P,X,j) = 0 and j <= q do j := j+1 od ;
  j ;
end proc ;
  description "Valuation du polynôme P éventuellement majorée par q
  (par ex si le polynôme est nul)"

voirHK := proc(L,m,n,p)
local i,j, L1 ;
  L1 := [seq(expand(L[i]), i=1..n+m-1)] ;
  if nargs=4 and <>p then
    L1 := [seq(normal(L1[i] mod p), i=1..n+m-1)]
  else L1 := [seq(normal(L1[i]), i=1..n+m-1)] fi ;
  matrix(m,n,[seq([seq(L1[i+j+1], j=0..n-1)],i=0..m-1)]) ;
end proc ;

```
    description "Creation d'une matrice de Hankel avec m lignes et n colonnes :
    L est une liste de taille au moins m+n-1 le tout éventuellement modulo premier"
VoirHK := proc(H,p)
local ta;
  ta := Taille(H);
  if nargs=1 then
    voirHK(H,ta,ta)
  else voirHK(H,ta,ta,p) fi;
end proc;
    description "Creation d'une matrice de Hankel de taille n codée par une liste
    de taille exactement 2n-1"
VoirTo := (L,p) -> evalm(Ara(Taille(L))*VoirHK(args));
end proc;
    description "Creation d'une matrice de Toeplitz de taille n codée
    par une liste de taille exactement 2n-1"
VoirToS := proc(L,p)
local i,n,LT;
  n := nops(L);
  LT := [seq(0,i=1..n-1),seq(L[i],i=1..n)];
  if nargs=1 then VoirTo(LT)
  else VoirTo(LT,p) fi;
end proc;
    description "Creation d'une Toeplitz supérieure, éventuellement modulo p"
```

A.2 Les procédures de diagonalisation par inversion de matrices Toeplitz supérieures

```
QuoResCroiss := proc(R0,R1,X::name,r,q,p)
local L0,L1,LQ,i,j,k,b,c ;
  L0:= [seq(coeff(R0,X,i),i=0..q)] ;
  L1:= [seq(coeff(R1,X,i),i=0..q)] ;
  LQ:= [seq(0,i=0..r)] ;
  if nargs=5 or p=0 then
    b:= 1/L1[1] ;
    for i from 1 to r+1 do
      c:= b*L0[i] ;
      LQ[i]:= c ;
      for j from i+1 to q+1 do
        L0[j]:= L0[j]-c*L1[j-i+1]
      od ;
    od
  else
    b:= 1/L1[1] mod p ;
    for i from 1 to r+1 do
      c:= b*L0[i] mod p ;
      LQ[i]:= c ;
      for j from i+1 to q+1 do
        L0[j]:= L0[j]-c*L1[j-i+1] mod p
      od ;
    od
  fi ;
  [LiPo(LQ,X),sum(L0[k]*X^(k-1),k=r+2..q+1)] ;
end proc ;
```

description "Quotient et Reste dans une division en puissance croissantes pour deux polynômes $R0$ et $R1$, le quotient Q sera de degré r les calculs sont effectués modulo $X^{(q+1)}$, $R1$ est supposé être de valuation 1, les calculs sont faits éventuellement modulo p, la sortie est [Q,R] avec Q le quotient et R le reste"

```
RMHCroiss := proc(L,X::name,p)
local S,i,m,n,r,s,q,R0,R1,R2,Q,QR,Stu ;
  Stu:= NULL ;S:= LiPo(L,X) ; n:= nops(L) ;
  if (nargs=3) and (p<>0) then S:= S mod p fi ;
  r:= 1+ValIDL(S,X,n) ;q:= n-r ; m:= iquo(n+1,2) ;
  R0:= 1 ;
  R1:= sum(coeff(S,X,i+r-1)*X^(i),i=0..q) ;
  while r <q to q/2 do
    QR:= QuoResCroiss(R0,R1,X,r,q,p) ;
    Q:= QR[1] ;
    R2:= QR[2] ;
    Stu:= Stu,[Q,r] ;
    s:= ValIDL(R2,X,q) ;
    q:= q-s ;
```

```
    R0 := ReDL(R1,X,q) ;
    R1 := sum(-coeff(R2,X,i+s)*X^i,i = 0..q) ;
    m := m-r ; r := s-r ;
  od ;
  if q> = 0 then
    QR := QuoResCroiss(R1,R0,X,q,q,p) ; Q := QR[1] ;
    Stu := Stu,[Q,m]
  else Stu := Stu,[0,m] fi ;
  [Stu] ;
end proc ;
    description "Réduction par division en puissances croissantes"
```

A.3 Les procédures de diagonalisation avec les divisions en puissances décroissantes

```
RedH := proc(H1,p)
local i,n,ta,r,T,S,b1,H,b ;
  b := evalb(nargs = 2 and p <> 0) ;
  if b then H := EVL(H1,p) else H := H1 fi ;
  n := nops(H) ; ta := Taille(H) ; b1 := false ;
  if n = 1 then RETURN([H,true])
    else
      r := 0 ; while H[r+1] = 0 and (r <ta) do r := r+1 od ;
      if r> = ta-1 then b1 := true ; T := H
      else
        S := [seq(H[i], i = r+1..n)] ;
        if b then T := InvToS(S,p) else T := InvToS(S) fi ;
        T := [seq(-T[i],i = r+3..n-r)]
      fi ;
    [T,b1]
  fi ;
end proc ;
```

description "On calcule les coefficients du deuxième bloc Hankel dans la réduite de la matrice de Hankel ayant pour coefficients H, on indique s'il n'y avait rien à réduire dans le booléen b1 qui est la deuxième composante de la sortie"

```
RedHQuo := proc (H1,p)
local i,n,ta,r,T,S,b1,H,Quo,b ;
  b := evalb(nargs = 2 and p <> 0) ;
  if b then H := EVL(H1,p) else H := H1 fi ;
  n := nops(H) ; ta := Taille(H) ; b1 := false ;
  if n = 1 then RETURN([[H],true])
    else
    r := 0 ; while H[r+1] = 0 and (r<ta) do r := r+1 od ;
    if r>= ta-1 then b1 := true ; T := H
      else
      S := [seq(H[i], i = r+1..n)] ;
      if b then T := InvToS(S,p) else T := InvToS(S) fi ;
      Quo := DebutListe(T,r+2) ;
      T := [seq(-T[i],i = r+3..n-r)]
    fi ;
    [[T,Quo],b1]
  fi ;
end proc ;
```

description "Comme le précédent mais on donne aussi le quotient en plus du reste"

```
PassRedH := proc(H1,p)
local i,n,ta,r,T,S,b1,H,Quo,b ;
  b := evalb(nargs = 2 and p <> 0) ;
  if b then H := EVL(H1,p) else H := H1 fi ;
```

```
  n:=nops(H) ; ta:=Taille(H) ; b1:=false ;
  if n=1 then RETURN([[H],true])
    else
    r:=0 ; while H[r+1]=0 and (r<ta) do r:=r+1 od ;
    if r >=ta-1 then b1:=true ; T:=H
      else
      S:=[seq(H[i], i=r+1..n)] ;
      if b then T:=InvToS(S,p) else T:=InvToS(S) fi ;
    fi ;
    VoirToS(DebutListe(T,ta))
  fi ;
end proc ;
```

description "Comme le précédent mais donne juste la matrice de passage"

```
IteRHQuo:=proc(H,p)
local C,b,SH,H1,b1,i,n ;
  b:=evalb(nargs=2 and p<>0) ; n:=nops(H) ;
  if b then H1:=[seq(H[i] mod p, i=1..n)]
    else H1:=H
  fi ;
  SH:=NULL ; b1:=false ;
  C:=[[H1],b1] ;
    to n while not b1 do
      SH:=SH,C[1] ;
      if b then C:=RedHQuo(C[1,1],p)
        else C:=RedHQuo(C[1,1])
      fi ;
      b1:=C[2] ;
    od ;
  [SH] ;
end proc ;
```

description "On itère **RedHQuo**, on retourne la liste des quotients et restes sous forme de listes"

```
IRHQ:=proc(H,p)
local i,n,SH ;
  SH:=IteRHQuo(args) ; n:=nops(SH) ;
  [seq(LiPo2(SH[i,2],X),i=2..n)],SH[n,1]
end proc ;
```

description "on itere RedHQuo, on retourne la liste des quotients sous forme de polynomes"

```
IRHR:=proc(H,p)
local i,SH ;
  SH :=IteRHQuo(args) ;
  [seq(SH[i,1],i=2..nops(SH))]
end proc ;
```

description "on itere RedHQuo, on retourne la liste des restes sous forme de listes"

A.4 Les procédures de diagonalisation avec les divisions en puissances décroissantes avec troncature

Tronk := proc(P,X::name,k) ;
  expand(evala(Quo(P,X^(k),X))) ;
end proc ;
  description "on retourne le quotient de la division de ($P(X)$-monômes de degré $< k$) par X^k "

Tronq := proc(P,X::name,k) ;
  expand(X^(k) evala(Quo(P,X^(k+1),X))) ;
end proc ;
  description "on retourne le quotient de la division de ($P(X)$-monômes de degré $\leq k$) par X "

QuoRemTron := proc(P,Q,X,p1)
local m, p, R0,R1,R2,QR1 ;
  if nargs = 3 then p := 0 else p := p1 fi ;
  R0 := P ; R1 := Q ;
  m := degre(R0,X)-degre(R1,X)-1 ;
  QR1 := **QuoRem**(R0,R1,X,p) ; R2 := -QR1[2] ;
  R1 := **Tronk**(R1,X,m) ;
  R2 := **Tronk**(R2,X,m) ;
  [QR1[1],R1,R2] ;
  end proc ;
  description "On retourne le quotient et le reste tronqués
  (éventuellement modulo p) de la division de A par B"

RMHDecTronq := proc(P,Q,X::name,p1)
local dR1,R0,R1,R2,QR,Stu,test,m,p,trestant,dR2,dR0,R3 ;
  if nargs = 3 then p := 0 else p := p1 fi ;
  R0 := P ; R1 := Q ; Stu := NULL ;
  dR0 := r(R0,X) ; dR1 := r(R1,X) ; test := 2*dR1-dR0 ;
  trestant := (dR0+1)/2 ;
  to degree(P,X) while test > 0 do
    QR := **QuoRemTron**(R0,R1,X,p) ;
    R1 := QR[2] ; R2 := QR[3] ; dR1 := r(R1,X) ; dR2 := r(R2,X) ;
    m := dR1-dR2 ; trestant := trestant-degree(QR[1],X) ;
    R0 := **Tronq**(R1,X,m) ;
    R1 := **Tronq**(R2,X,0) ; dR0 := r(R0,X) ; dR1 := r(R1,X) ;
    test := 2*dR1-dR0 ;
    Stu := Stu,sort(QR[1],X) ;
  od ;
  if dR1 >= 0 then
    R1 := X^(dR1)*R1 ;
    R0 := **Tronk**(R0,X,dR0-dR1) ;
    QR := **QuoRem**(R1,R0,X,p) ;
    Stu := Stu,sort(QR[1],X),[trestant] ;
  else

```
    Stu := Stu,0,[trestant] ;
  fi ;
  eval(Stu) ;
end proc ;
```

description "Dans cette procédure $P := X^{2n-1}$, $Q :=$ sort(LiPo2(S, X)) où S est la liste de taille $2n-1$ qui code la matrice de Hankel d'ordre n."

Annexe B. Quelques capture écran

Il s'agit dans ce chapitre de mettre en œuvre les procédures correspondantes au chapitre précédent.
Pour cela considérons la liste :

$$H := [34, -12, 28, 2, -16, 41, 18, -41, -40, -23, 32, 6, -15, 2, -22, -21, -45, 11, 50, -15, -32, -5, 30, 18, -9, 40, 35, 1, -31],$$

Sans perte de généralité tous les calcul ci-après ce fait modulo le nombre premier $p = 101$.

Avec **RMHCroiss**(H, X, p) :

On retourne la liste des quotients qui codent les blocs diagonaux Hankel-inférieurs (et leurs ordres respectifs) de la réduite (Fig.B.1) :

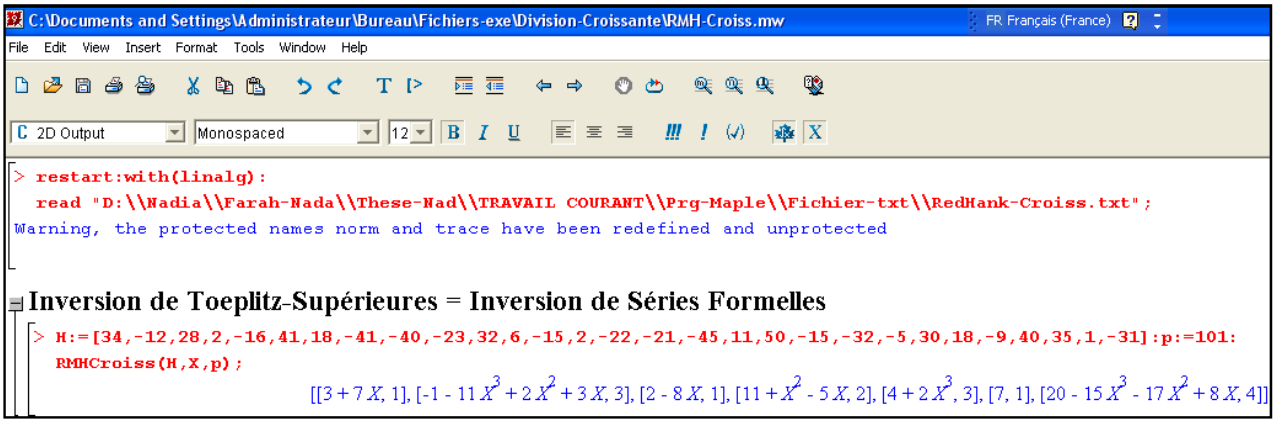

Figure B.1 – divisions en puissances croissantes de polynômes

Avec **IRHQ**(H, p) :

On retourne la liste des quotients qui codent les blocs diagonaux Hankel-inférieurs de la réduite (Fig.B.2) : :

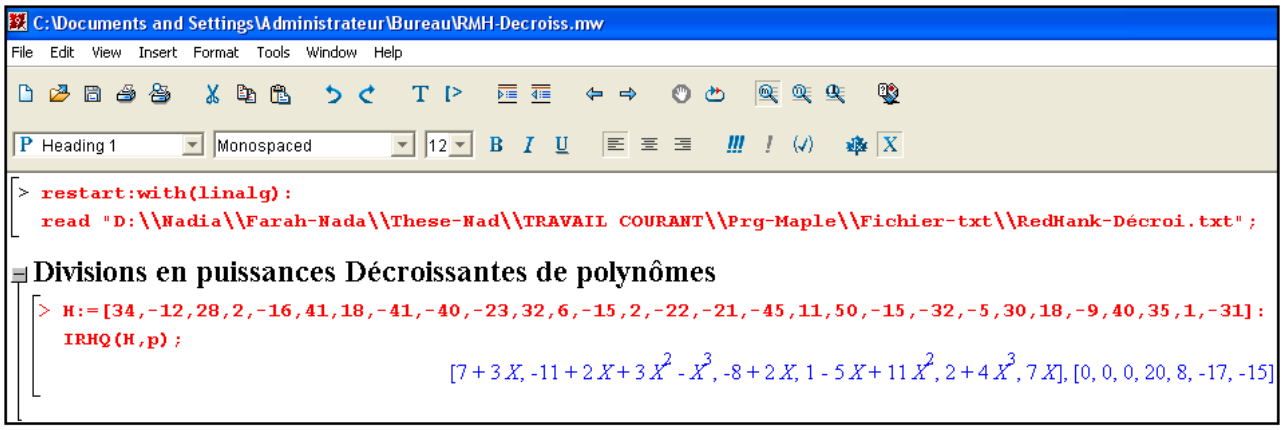

Figure B.2 – divisions en puissances décroissantes de polynômes

Avec **RMHDecTronq**(P, Q, X, p) :

Ici on a :

$$Q(X) = 34X^{28} - 12X^{27} + 28X^{26} + 2X^{25} - 16X^{24} + 41X^{23} + 18X^{22} - 41X^{21} - 40X^{20} - 23X^{19} + 32X^{18} + 6X^{17} - 15X^{16} + 2X^{15} - 22X^{14} - 21X^{13} - 45X^{12} + 11X^{11} + 50X^{10} - 15X^9 - 32X^8 - 5X^7 + 30X^6 + 18X^5 - 9X^4 + 40X^3 + 35X^2 + X - 31,$$

et $P(X) = X^{29}$.
On retourne la liste des quotients qui codent les blocs diagonaux Hankel-inférieurs (et leurs ordres respectifs) (Fig.B.3) :

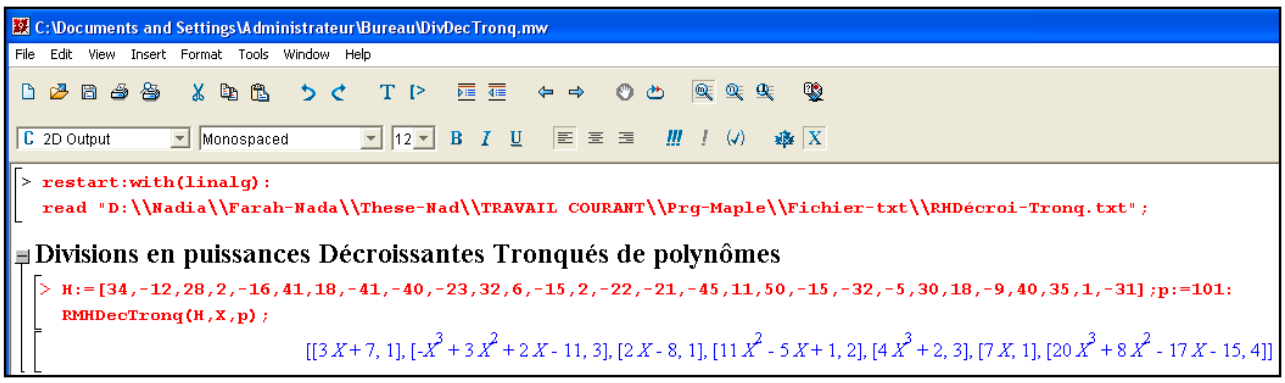

Figure B.3 – divisions en puissances décroissantes de polynômes avec troncature

Remarquons que ces quotients sont les réciproques de ceux retournés par **RMHCroiss**.

Comparaison avec la méthode classique : avec **RedComHNed**(H, p)

Avec la procédure `RedComHNed` qui prend en entrée la liste H qui code la matrice de Hankel h et qui donne en sortie la matrice de passage A et la réduite diagonale par bloc Hankel-inférieurs D, on aura :

`RedComHNed`$(H, 101)$:

$$\mathrm{h} = \begin{pmatrix}
34 & -12 & 28 & 2 & -16 & 41 & 18 & -41 & -40 & -23 & 32 & 6 & -15 & 2 & -22 \\
-12 & 28 & 2 & -16 & 41 & 18 & -41 & -40 & -23 & 32 & 6 & -15 & 2 & -22 & -21 \\
28 & 2 & -16 & 41 & 18 & -41 & -40 & -23 & 32 & 6 & -15 & 2 & -22 & -21 & -45 \\
2 & -16 & 41 & 18 & -41 & -40 & -23 & 32 & 6 & -15 & 2 & -22 & -21 & -45 & 11 \\
-16 & 41 & 18 & -41 & -40 & -23 & 32 & 6 & -15 & 2 & -22 & -21 & -45 & 11 & 50 \\
41 & 18 & -41 & -40 & -23 & 32 & 6 & -15 & 2 & -22 & -21 & -45 & 11 & 50 & -15 \\
18 & -41 & -40 & -23 & 32 & 6 & -15 & 2 & -22 & -21 & -45 & 11 & 50 & -15 & -32 \\
-41 & -40 & -23 & 32 & 6 & -15 & 2 & -22 & -21 & -45 & 11 & 50 & -15 & -32 & -5 \\
-40 & -23 & 32 & 6 & -15 & 2 & -22 & -21 & -45 & 11 & 50 & -15 & -32 & -5 & 30 \\
-23 & 32 & 6 & -15 & 2 & -22 & -21 & -45 & 11 & 50 & -15 & -32 & -5 & 30 & 18 \\
32 & 6 & -15 & 2 & -22 & -21 & -45 & 11 & 50 & -15 & -32 & -5 & 30 & 18 & -9 \\
6 & -15 & 2 & -22 & -21 & -45 & 11 & 50 & -15 & -32 & -5 & 30 & 18 & -9 & 40 \\
-15 & 2 & -22 & -21 & -45 & 11 & 50 & -15 & -32 & -5 & 30 & 18 & -9 & 40 & 35 \\
2 & -22 & -21 & -45 & 11 & 50 & -15 & -32 & -5 & 30 & 18 & -9 & 40 & 35 & 1 \\
-22 & -21 & -45 & 11 & 50 & -15 & -32 & -5 & 30 & 18 & -9 & 40 & 35 & 1 & -31
\end{pmatrix}$$

dont les mineurs principaux dominants : [34, 0, 0, −26, 21, 0, −24, 0, 0, −49, −18, 0, 0, 0, −29]

Les blocs seront d'ordre respectif : [1, 3, 1, 2, 3, 1, 4]

Notre matrice de PASSAGE est :

$$A = \begin{pmatrix}
3 & -7 & 21 & 14 & 46 & 20 & 46 & -48 & 0 & 0 & -43 & 12 & 0 & 0 & 0 \\
0 & -3 & 2 & 27 & -38 & 24 & -46 & 30 & -48 & 0 & -48 & 0 & 12 & 0 & 0 \\
0 & 0 & -3 & 2 & -47 & 34 & -19 & 5 & 30 & -48 & 28 & -24 & 0 & 12 & 0 \\
0 & 0 & 0 & -3 & 4 & 14 & 46 & -33 & 5 & 30 & 40 & 11 & -24 & 0 & 12 \\
0 & 0 & 0 & 0 & -6 & 5 & -25 & 5 & -33 & 5 & -19 & -37 & 11 & -24 & 0 \\
0 & 0 & 0 & 0 & 0 & 35 & 35 & -23 & 5 & -33 & 47 & 12 & -37 & 11 & -24 \\
0 & 0 & 0 & 0 & 0 & 0 & 35 & 39 & -23 & 5 & -44 & 12 & 12 & -37 & 11 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 39 & 39 & -23 & 20 & 22 & 12 & 12 & -37 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 39 & 39 & 41 & 20 & 22 & 12 & 12 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 39 & -30 & 41 & 20 & 22 & 12 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & 41 & 20 & 22 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & 41 & 20 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & 41 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30 & -30 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30
\end{pmatrix}$$

et notre RÉDUITE diagonale par bloc Hankel-inférieurs est :

$$D = {}^{t}A\,h\,A = \begin{pmatrix}
3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 8 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 8 & -17 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 8 & -17 & -15 & 0
\end{pmatrix}$$

Avec la méthode classique de LU-décomposition : Matrice de PASSAGE à Diagonale unitaire

$$A_1 = \begin{pmatrix}
1 & 36 & -7 & 29 & 26 & 15 & -16 & -9 & 0 & 0 & 25 & 40 & 0 & 0 & 0 \\
0 & 1 & 33 & -9 & 40 & 18 & 16 & -7 & -9 & 0 & 42 & 0 & 40 & 0 & 0 \\
0 & 0 & 1 & 33 & -9 & -25 & 11 & -18 & -7 & -9 & 26 & 21 & 0 & 40 & 0 \\
0 & 0 & 0 & 1 & 33 & -40 & -16 & 38 & -18 & -7 & -35 & 3 & 21 & 0 & 40 \\
0 & 0 & 0 & 0 & 1 & 29 & -44 & -18 & 38 & -18 & 4 & 45 & 3 & 21 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & -18 & 38 & 22 & 40 & 45 & 3 & 21 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & -18 & -12 & 40 & 40 & 45 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 & 40 & 45 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 & 40 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

Dans la méthode classique, le premier bloc reste inchangé

$$D_1 = {}^{t}A_1 h A_1 = \begin{pmatrix}
34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -45 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -45 & 34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -45 & 34 & -11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -38 & -47 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -33 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -33 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 44 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 44 & -43 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 44 & -43 & -32
\end{pmatrix}$$

Matrice de PASSAGE CLASSIQUE

$$R = \begin{pmatrix}
1 & 36 & 0 & 0 & 26 & 15 & 0 & -9 & 0 & 0 & 25 & 40 & 0 & 0 & 0 \\
0 & 1 & 36 & 0 & 40 & 18 & 15 & -7 & -9 & 0 & 42 & 0 & 40 & 0 & 0 \\
0 & 0 & 1 & 36 & -9 & -25 & 18 & -18 & -7 & -9 & 26 & 21 & 0 & 40 & 0 \\
0 & 0 & 0 & 1 & 33 & -40 & -25 & 38 & -18 & -7 & -35 & 3 & 21 & 0 & 40 \\
0 & 0 & 0 & 0 & 1 & 29 & -40 & -18 & 38 & -18 & 4 & 45 & 3 & 21 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 29 & 2 & -18 & 38 & 22 & 40 & 45 & 3 & 21 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & -18 & -12 & 40 & 40 & 45 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 & 40 & 45 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 & 40 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 & 40 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 33 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

RÉDUITE CLASSIQUE

$$D_R = {}^{t}RhR = \begin{pmatrix}
34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -45 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -45 & -34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -45 & -34 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -38 & 47 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -33 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -33 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 44 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 44 & -43 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 44 & -43 & -32
\end{pmatrix}$$

Bibliographie

[1] J. Abdeljaoued and H. Lombardi. *Méthodes Matricielles. Introduction à la Complexité Algébrique*. Springer, collection Mathématiques et Applications de la SMAI, (2003). 3, 67

[2] S. Barnett. *Polynomials and Linear Control Systems*, Marcel Dekker, (1983). 45

[3] S. Basu, R. Pollack and M.F. Roy. *Algorithms in real algebraic geometry*. Springer, (2003). 7, 43, 57, 69

[4] N. Ben Atti and G.M. Diaz–Toca. *Block diagonalization and LU-equivalence of Hankel matrices*, Linear Algebra and its Applications **412**, 247-269 (2006). 7, 57

[5] N. Ben Atti , G.M. Diaz–Toca and H. Lombardi. *The Berlekamp–Massey Algorithm revisited*, Applicable Algebra in Engineering, Comunication and Computing 17,1, 75-82(2006). 63, 65

[6] E.R. Berlekamp. *Algebraic Coding Theory*, McGraw-Hill, New York, ch.7 (1968). 64

[7] D. Bini and L. Gemignani. *Fast Parallel Computations of the Polynomial Remainder Sequence via Bezout and Hankel Matrices*, SIAM J.Comput, Vol. 24, No. 1, 63-77, February (1995). 2, 7, 41, 44, 55

[8] D. Bini and V. Pan. *Polynomial and Matrix Computations. Volume 1 Fundamental Algorithms*. Birkhäuser (1994). 7, 13, 41, 43, 44

[9] A. Bultheel and M. Van Barel. *Linear Algebra, rational approximation and orthogonal polynomials*. Studies in Computational Mathematics 6, Elsevier/North-Holland, Amsterdam, December (1997). 7, 33

[10] U. Cheng. *On the continued fraction and Berlekamp's algorithm*, IEEE Trans. Inform. Theory, vol. IT-30, 541–44 (1984). 65

[11] J. Della Dora, C. Dicrescenzo and Duval D. *About a new method for computing in algebraic number fields*. In Caviness B.F. (Ed.) EUROCAL '85. Lecture Notes in Computer Science 204, 289–290. Springer (1985). 73

[12] G.M. Diaz–Toca and L. Gonzalez–Vega. *Barnett's Theorem about the greatest common divisor of several univariate polynomials through Bezout–like Matrices*. Journal of Symbolic Computation **34**, 1, 59–81 (2002).

[13] G.M. Diaz–Toca and L. Gonzalez–Vega. *Various New Expressions for Subresultants and Their Appplications*. Applicable Algebra in Engineering, Communication and Computing (AAECC) **15**, 3–4, 233–266, (2004). 43

[14] J.L. Dornstetter. *On the equivalence Between Berlekamp's and Euclid's Algorithm*, IEEE Trans. Inform. Theory, **33**/3,428–431 (1987). 43
64, 65, 75

[15] F.G. Frobenius. *Uberdas Traegheitsgesetzdes quadratishen Formen*, J. ReineAngew. Math, **114**,187-230 (1895). 44

[16] P.A. Fuhrmann. *A Polynomial Approch to Linear Algebra*. Springer, (2000). 37, 42

[17] F. Gantmacher. *Théorie des matrices, Tome 1, (Théorie Générale)*, Dunod, Paris, (1966). 37

[18] J. von zur Gathen and T. Lücking. *Subresultants revisited*. Theoretical Computer Science **297**, 199–239 (2003).

[19] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*, Cambridge University Press (2003). 64

[20] L. Gemignani. *Computationally Efficient Applications of the Euclidean algorithm to Zero Location*. Linear Algebra and its Applications **249**, 79–91 (1996). 57

[21] E.R. Berlekamp. *Binary BCH Codes for Correcting Multiple Errors*, Key paper in the Development of Coding Theory, IEEE Press, 145–148 (1974). 64

[22] W.B. Gragg and A. Lindquist. *On the partial realization problem*. Linear Algebra Appl. **50**, 277–319 (1983). 7, 33

[23] G. Heinig and K. Rost. *Algebraic Methods for Toeplitz-like Matrices and Operators*. Birkhäuser Verlag, Basel. (1984). 7

[24] U. Helmke and P.A. Fuhrmann, *Bezoutians*, Linear Algebr. Appl., **122/123/124** , 1039–1097 (1989). 43

[25] E. Jonckheere and C. Ma. *A simple Hankel Interpretation of the Berlekamp–Massey Algorith*, Linear Algebra and its Applications **125**, 65–76 (1989). 65

[26] P. Lancaster and M. Tismenetsky. *The Theory of Matrices With Applications*. Academic Press ; 2 edition (1985). 41, 57

[27] T. Lickteig and M.F. Roy. *Sylvester–Habicht Sequences and Fast Cauchy Index Computation*. J. Symbolic Computation **31**, 315–341 (2001). 57

[28] J.L. Massey. *Shift register synthesis and BCH decoding*, IEEE Trans. Inform. Theory, vol. IT-15, 122–127 (1969). 64

[29] M. Mignotte. *Mathematics for Computer Algebra*, Universitext, Springer (1992). 45

[30] W.H. Mills. *Continued Fractions and Linear Recurrences*, Math. Comput. 29, 173–180 (1975). 65

[31] V. Pan. *New Techniques for the Computation of linear recurrence coefficients*, Finite Fields and Their Applications **6**, 93–118 (2000). 65

[32] V. Shoup. *A Computational Introduction to Number Theory and Algebra*, Cambridge University Press (2005). 65

[33] Y. Sugiyama and al. *A method for solving key equation for decoding Goppa codes*, Infor. Contr. vol 27, 87–99 (1975). 65

[34] L.R. Welch and R.A. Scholtz. *Continued fractions and Berlekamp's algorithm*, IEEE Trans. Inform. Theory, vol. IT-25, 18–27 (1979). 65

RÉSUMÉ

Cette thèse présente une contribution à l'amélioration de certains résultats concernant les algorithmes en Algèbre linéaire et plus particulièrement les algorithmes sur les matrices structurées.

Nous présentons un nouvel algorithme de diagonalisation par blocs des matrices de Hankel, particulièrement efficace.

Dans le cas où la matrice de Hankel correspond à une suite récurrente linéaire, nous retrouvons ainsi l'algorithme de Berlekamp-Massey, mais dans une version simplifiée (plus facile à expliquer et à programmer) et accélérée par des troncatures.

En outre notre version permet une gestion dynamique des données.

Notre diagonalisation par blocs, qui s'applique sur un corps arbitraire, nous permet de donner une démonstration purement algébrique simple d'un délicat théorème de Frobenius pour la signature d'une forme de Hankel réelle.

Nous donnons également une étude approfondie de l'algorithme d'Euclide signé et de ses versions matricielles pour les matrices de Hankel et de Bezout associées à un couple de polynômes. Nous expliquons les rapports existants entre différents algorithmes connus dans la littérature.

MOTS CLÉS : matrice de Bezout, matrice de Hankel, matrice de Toeplitz, suite récurrente linéaire, diagonalisation par blocs, Algorithme d'Euclide signé, Algorithme de Berlekamp-Massey.

AMS classification : 15A03 ; 15A15 ; 15A23 ; 15A24 ; 15A63 ; 47B35 ; 65F30 ; 68W30.

ABSTRACT

We give a new algorithm for the blocs diagonalization of Hankel matrices.

When the matrix corresponds to a linear recurrent sequence we obtain a simplified version of Berlekamp-Massey's algorithm, more easy to implement and to understand than the usual one. Also our version is slightly faster and allows us to use lazzy techniques.

We give a very simple and purely algebraic proof of the frobenius's theorem for the signature of real Hankel matrices.

We give a study of the Extended Euclidean Algorithm and of his matricial versions using Hankel or Bezout matrices. We explain the links between all these versions.

KEY WORDS : Bezout matrix, Hankel matrix, Toeplitz matrix, Linearly recurrent sequences, Block LU factorization, Euclidean algorithm, Berlekamp-Massey Algorithm.